Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision
UMR CNRS 7024

ANNALES DU LAMSADE N°2
Juin 2004

Numéro publié grâce au Bonus Qualité Recherche
accordé par l'Université Paris IX - Dauphine

Pour se procurer l'ouvrage, contactez Mme D. François, tél. 01 44 05 42 87
e-mail : francois@lamsade.dauphine.fr

COLLECTION « CAHIERS, DOCUMENTS ET NOTES » DU LAMSADE

La collection « Cahiers, Documents et Notes » du LAMSADE publie, en anglais ou en français, des travaux effectués par les chercheurs du laboratoire éventuellement en collaboration avec des chercheurs externes. Ces textes peuvent ensuite être soumis pour publication dans des revues internationales. Si un texte publié dans la collection a fait l'objet d'une communication à un congrès, ceci doit être alors mentionné. La collection est animée par un comité de rédaction.

Toute proposition de cahier de recherche est soumise au comité de rédaction qui la transmet à des relecteurs anonymes. Les documents et notes de recherche sont également transmis au comité de rédaction, mais ils sont publiés sans relecture. Pour toute publication dans la collection, les opinions émises n'engagent que les auteurs de la publication.

Depuis mars 2002, les cahiers, documents et notes de recherche sont en ligne. Les numéros antérieurs à mars 2002 peuvent être consultés à la Bibliothèque du LAMSADE ou être demandés directement à leurs auteurs.

Deux éditions « papier » par an, intitulées « Annales du LAMSADE » sont prévues. Elles peuvent être thématiques ou représentatives de travaux récents effectués au laboratoire.

COLLECTION "CAHIERS, DOCUMENTS ET NOTES" OF LAMSADE

The collection “Cahiers, Documents et Notes” of LAMSADE publishes, in English or in French, research works performed by the Laboratory research staff, possibly in collaboration with external researchers. Such papers can (and we encourage to) be submitted for publication in international scientific journals. In the case one of the texts submitted to the collection has already been presented in a conference, it has to be mentioned. The collection is coordinated by an editorial board.

Any submission to be published as “cahier” of LAMSADE, is sent to the editorial board and is refereed by one or two anonymous referees. The “notes” and “documents” are also submitted to the editorial board, but they are published without refereeing process. For any publication in the collection, the authors are the unique responsible for the opinions expressed.

Since March 2002, the collection is on-line. Old volumes (before March 2002) can be found at the LAMSADE library or can be asked directly to the authors.

Two paper volumes, called “Annals of LAMSADE” are planned per year. They can be thematic or representative of the research recently performed in the laboratory.

Foreword

This is the second volume of the Annals of the LAMSADE. As the former one, it is multi-thematic. Accepted papers are a sample of either the research works performed in the laboratory, or of the advances in its research domains.

We publish here two contributions of researchers who, even if they are not part of the LAMSADE's staff, are close collaborators for many years. The first paper is a survey on the complexity and the efficiency of local search methods. The second one, presents a new database management algorithm and studies its complexity. We hope that in the volumes to come, we will be able to publish more contributions issued from our external partners.

For this volume, the editorial board has encouraged papers co-authored by PhD students of the laboratory. This is the case of seven among the eleven papers accepted.

Vangelis Th. PASCHOS
Editor-in-Chief

Editorial

Ce deuxième numéro des Annales du LAMSADE est, comme le précédent, pluri-thématique. Les articles publiés reflètent soit les travaux de recherche effectués dans le laboratoire, soit des avancées dans les domaines de recherche du laboratoire.

Par ailleurs, nous accueillons deux articles de chercheurs qui, même s'ils ne sont pas des membres permanents du LAMSADE, travaillent étroitement avec nous depuis longtemps. Le premier est un tour d'horizon sur la complexité et l'efficience des méthodes de la recherche locale. Le deuxième propose un nouvel algorithme de gestion des bases de données et analyse sa complexité. Nous espérons que dans l'avenir, nous allons accueillir plus d'articles issus de nos partenaires externes.

Pour ce deuxième numéro des Annales, le comité de rédaction a suscité des articles co-écrits par des étudiants en thèse au LAMSADE. Une large place leur a été consacrée, sept d'entre eux figurent au sommaire des onze articles présentés dans ce volume.

Vangelis Th. PASCHOS
Rédacteur en chef

Sommaire/Contents

Annales du LAMSADE N°2

PTAS-completeness in standard and differential approximation

Cristina Bazgan[*], Bruno Escoffier[*], Vangelis Th. Paschos[*]

Résumé

Nous nous plaçons dans le cadre de l'approximation polynomiale des problèmes d'optimisation. Les réductions préservant l'approximabilité ont permis de structurer les classes d'approximation classiques (**APX**, **PTAS**,...) en introduisant des notions de complétude. Par exemple, des problèmes naturels ont été montrés **APX**- ou **DAPX**-complets (pour le paradigme de l'approximation différentielle), sous des réductions préservant l'existence de schémas d'approximation polynomiaux. Nous introduisons ici une notion de **PTAS**-complétude pour laquelle des problèmes naturels sont **PTAS**-complets. Nous définissons également une notion analogue de **DPTAS**-complétude pour l'approximation différentielle, et montrons l'existence de problèmes **DPTAS**-complets naturels. Ensuite, nous étudions l'existence de problèmes intermédiaires (sous nos réductions) et répondons partiellement à la question en montrant que l'existence de problème **NPO**-intermediaires sous la réduction de Turing est une condition suffisante. Enfin, nous montrons que MIN COLORING est **DAPX**-complet sous la DPTAS-réduction (définie dans "G. Ausiello, C. Bazgan, M. Demange, et V. Th. Paschos, *Completeness in differential approximation classes*, MFCS'03").

Mots-clefs : Algorithme approché, complétude, complexité, optimisation combinatoire, schéma d'approximation, réduction.

Abstract

This article focuses on polynomial approximation of optimization problems. The classical approximation classes (**APX**, **PTAS**,. . .) have been structured by the introduction of approximation-preserving reductions and notions of completeness. For instance, natural problems are known to be **APX**- or **DAPX**-complete (under the

[*] LAMSADE, Université Paris-Dauphine, 75775 Paris cedex 16, France. {bazgan,escoffier,paschos}@lamsade.dauphine.fr

differential approximation paradigm), under suitably defined reductions preserving polynomial time approximation schemata. We introduce here a notion of **PTAS**-completeness for which natural problems are shown to be **PTAS**-complete. We also define an analogous notion of **DPTAS**-completeness for the differential approximation, and show the existence of natural **DPTAS**-complete problems. Next, we deal with the existence of intermediate problems, under our reductions and we partially answer this question showing that the existence of **NPO**-intermediate problems under Turing-reduction is a sufficient condition. Finally, we show that MIN COLORING is **DAPX**-complete under DPTAS-reduction (defined in "G. Ausiello, C. Bazgan, M. Demange, and V. Th. Paschos, *Completeness in differential approximation classes*, MFCS'03").

Key words : Approximation algorithm, approximation schema, combinatorial optimization, completeness, complexity, reduction.

1 Introduction

Many **NP**-complete problems are decision versions of natural optimization problems. Since, unless **P** = **NP**, such problems cannot be solved in polynomial time, a major question is to find polynomial algorithms producing solutions "close to the optimum" (in some prespecified sense). Here, we deal with polynomial approximation of **NPO** problems, i.e., for optimization problems the decision versions of which are in **NP**. A polynomial approximation algorithm A for an optimization problem Π is a polynomial time algorithm that produces, for any instance x of Π, a feasible solution $y = \mathtt{A}(x)$. The quality of y is estimated by computing the so-called approximation ratio. Two approximation ratios are commonly used in order to evaluate the approximation capacity of an algorithm: the standard ratio and the differential ratio.

By means of these ratios, **NPO** problems are then classified with respect to their approximability properties. Particularly interesting approximation classes are, for the standard approximation paradigm, class **APX** (the class of constant-approximable problems), **PTAS** (the class of problems admitting polynomial time approximation schemata) and **FPTAS** (the class of problems admitting fully polynomial time approximation schemata). Analogous classes can be defined under the differential approximation paradigm: **DAPX**, **DPTAS** and **DFPTAS** (see section 2 for formal definitions), are the differential counterparts of **APX**, **PTAS** and **FPTAS**, respectively. Note that **FPTAS** $\subsetneq$ **PTAS** $\subsetneq$ **APX**, and **DFPTAS** $\subsetneq$ **DPTAS** $\subsetneq$ **DAPX**; these inclusions are strict unless **P** = **NP**.

During last two decades, several approximation-preserving reductions have been introduced. Via them, hardness results in several approximability classes have been studied. Consider two classes $\mathbf{C_1}$ and $\mathbf{C_2}$ with $\mathbf{C_1} \subseteq \mathbf{C_2}$, and assume a reduction preserving membership in $\mathbf{C_1}$ (i.e., if Π reduces to Π' and $\Pi' \in \mathbf{C_1}$, then $\Pi \in \mathbf{C_1}$). A problem $\mathbf{C_2}$-complete under this reduction is in $\mathbf{C_1}$ iff $\mathbf{C_2} = \mathbf{C_1}$ (as in the case of **NP**-completeness).

Consider, for instance, the P-reduction in [5]; this reduction, extended in [3, 6] (and renamed PTAS-reduction), preserves membership in **PTAS**. Natural problems, such as MAX INDEPENDENT SET IN BOUNDED DEGREE GRAPHS ([13]) or MIN METRIC TSP ([14]), are **APX**-complete under the PTAS-reduction. This implies that such problems are not in **PTAS** unless $\mathbf{P} = \mathbf{NP}$ (since, as we have previously mentioned, provided that $\mathbf{P} \neq \mathbf{NP}$, then $\mathbf{PTAS} \subsetneq \mathbf{APX}$).

In differential approximation, analogous results have been obtained in [1] for differential approximation; there, a DPTAS-reduction, preserving membership in **DPTAS**, is defined and natural problems such as MAX INDEPENDENT SET IN BOUNDED DEGREE GRAPHS are shown to be **DAPX**-complete.

In the same way, the F-reduction of [5], preserves membership in **FPTAS**. Under this reduction, only one (not very natural) problem (derived from MAX VARIABLE-WEIGHTED SAT) is known to be **PTAS**-complete (see Appendix A). Despite some restrictive notions of **DPTAS**-hardness presented in [1], no systematic study of **DPTAS**-completeness has been performed until now.

Reductions provide a structure in approximation classes, and are very useful in obtaining hardness approximability results. As in the case of **NP**-completeness with the result of [12], one can try to refine the study of this structure by determining if there exist intermediate problems. For two complexity classes **C** and $\mathbf{C}'$, $\mathbf{C}' \subseteq \mathbf{C}$, and a reduction R preserving membership in $\mathbf{C}'$, a problem is called **C**-intermediate, if it is neither **C**-complete under R, nor in $\mathbf{C}'$. In [5] is proved the existence of **APX**- and **PTAS**-intermediate problems under P- and F-reductions, respectively.

We propose here a reduction preserving membership in **FPTAS**, weaker than the F-reduction of [5], for which natural problems are shown **PTAS**-complete. We also propose a reduction preserving membership in **DFPTAS** and show that, under it, natural problems as MIN VERTEX COVER, or MAX INDEPENDENT SET, both in planar graphs, are **DPTAS**-complete. Indeed, we show that, under our reduction, any polynomially bounded **NP**-hard problem of **PTAS** is **PTAS**-complete. Using another notion of polynomial boundness, diameter polynomial boundness, we show that any diameter polynomially bounded **NP**-hard problem of **DPTAS** is **DPTAS**-complete. Finally, we try to apprehend if our reductions allow existence of intermediate problems. We partially answer this question by proving that such problems do exist provided that there exist intermediate problems in **NPO** under the seminal Turing-reduction.

The paper is organized as follows: in Section 2, we recall some basic definitions and present our two reductions. In Sections 3 and 4, we present our completeness results in **PTAS** and **DPTAS**. The results on intermediate problems are given in Section 5. Finally, in Section 6, it is proved that MIN COLORING is **DAPX**-complete under DP-

TAS-reduction. This is the first problem that is **DAPX**-complete but not **APX**-complete. Definitions of problems used and/or discussed in the paper, together with specifications of their worst solutions are given in Appendix A.

2 Preliminaries

2.1 Polynomial approximation

We firstly recall some useful definitions about basic concepts of polynomial approximation.

Definition 1. A problem Π in **NPO** is a quadruple $(\mathcal{I}, \mathrm{Sol}, m, \mathrm{opt})$ where:

- $\mathcal{I}$ is the set of instances (and can be recognized in polynomial time);
- given $x \in \mathcal{I}$, $\mathrm{Sol}(x)$ is the set of feasible solutions of x; the size of a feasible solution of x is polynomial in the size $|x|$ of the instance; moreover, one can determine in polynomial time if a solution is feasible or not;
- Given $x \in \mathcal{I}$ and $y \in \mathrm{Sol}(x)$, $m(x, y)$ denotes the value of the solution y of the instance x; m is called the objective function, and is computable in polynomial time; we suppose here that $m(x, y) \in \mathbb{N}$;
- $\mathrm{opt} \in \{\min, \max\}$. ▮

Given a problem Π in **NPO**, we distinguish the following three different versions of it:

- the constructive version denoted also by Π, where the goal is to determine a solution $y^* \in \mathrm{Sol}(x)$ satisfying $m(x, y^*) = \mathrm{opt}\{v(x, y), y \in \mathrm{Sol}(x)\}$;
- the evaluation problem Π_e, where we are only interested in determining the value of an optimal solution;
- the decision version Π_d of Π where, given an instance x of Π and an integer k, we wish to answer the following question: "does there exist a feasible solution y of x such that $m(x, y) \geqslant k$ if $\mathrm{opt} = \max$, or $m(x, y) \leqslant k$ if $\mathrm{opt} = \min$?".

Given an instance x of an optimization problem Π, let $\mathrm{opt}(x)$ be the value of an optimal solution, and $\omega(x)$ be the value of a worst feasible solution. In other terms, $\omega(x)$ is the optimal value of the same optimization problem with the opposite objective (minimize instead of maximize, and vice-versa) with respect to Π. We now define the two ratios the most commonly used for the analysis of approximation algorithms, called standard and differential in the sequel.

Definition 2. Let x be an instance of a problem Π and $y \in \mathrm{Sol}(x)$. The standard approximation ratio of y is $r(x,y) = m(x,y)/\operatorname{opt}(x)$. The differential approximation ratio of y is $\delta(x,y) = |m(x,y) - \omega(x)|/|\operatorname{opt}(x) - \omega(x)|$. ▮

Following Definition 2, standard approximation ratios for minimization problems are greater than, or equal to, 1, while for maximization problems these ratios are smaller than, or equal to 1. On the other hand, differential approximation ratio is always at most 1 for any problem.

Let g be a function mapping the instances of a problem Π to $[0,1]$, or to $[1,+\infty)$. An algorithm A guarantees standard (resp., differential) ratio g iff, for any instance x of Π, $r(x,A(x)) \geqslant g(x)$, or $r(x,A(x)) \leqslant g(x)$, depending whether Π is a maximization or a minimization problem (resp., $\delta(x,A(x)) \geqslant g(x)$). A problem Π is standard (resp., differential) g-approximable iff there exists a polynomial algorithm that guarantees standard (resp., the differential) ratio g.

We can now formally define the approximation classes **APX**, **PTAS** and **FPTAS**. An **NPO** problem Π is in the class:

- **APX**, iff it is constant-approximable, i.e., iff there exists a polynomial algorithm which guarantees g where g does not depend on the instance;
- **PTAS**, iff it admits a polynomial time approximation schema; such a schema is a family of polynomial algorithms $\mathtt{A}_\varepsilon$, $\varepsilon \in]0,1]$, any of them guaranteeing approximation ratio $1-\varepsilon$, or $1+\varepsilon$;
- **FPTAS**, iff it admits a fully polynomial time approximation schema; such a schema is a polynomial time approximation schema $(\mathtt{A}_\varepsilon)_{\varepsilon\in]0,1]}$, where the complexity of any $\mathtt{A}_\varepsilon$ is polynomial in both the size of the instance and in $1/\varepsilon$.

Classes **DAPX**, **DPTAS** and **DFPTAS** for the differential approximation paradigm can be defined analogouslys.

An **NPO** problem Π is polynomially bounded iff there exists a polynomial q such that, for any instance x and for any feasible solution $y \in \mathrm{Sol}(x)$, $m(x,y) \leqslant q(|x|)$. It is diameter polynomially bounded iff there exists a polynomial q such that, for any instance x, $|\operatorname{opt}(x) - \omega(x)| \leqslant q(|x|)$. The class of polynomially bounded **NPO** problems will be denoted by **NPO-PB**, while the class of diameter polynomially bounded **NPO** problems will be denoted by **NPO-DPB**.

2.2 Reductions

First, let us recall that, given a reduction R and a set **C** of problems, a problem $\Pi \in \mathbf{C}$ is **C**-complete under R iff any problem in **C** R-reduces to Π. If R preserves membership in $\mathbf{C}' \subseteq \mathbf{C}$, Π is **C**-intermediate under R iff it is neither **C**-complete nor in $\mathbf{C}'$ (provided that $\mathbf{P} \neq \mathbf{NP}$). Moreover, we will say that a problem in **NPO** is **NP**-hard if its decision version is **NP**-complete.

In this paper, we will use three reductions. The first one is the seminal Turing-reduction between optimization problems as it appears in [9]. Turing-reduction only preserves optimality of solutions (and hence membership in $\mathbf{PO} \subseteq \mathbf{NPO}$ of polynomial time solvable problems).

Definition 3. Let Π and Π' be two problems in **NPO**. Then, Π reduces to Π' under Turing-reduction (denoted by $\Pi \leq_{\mathsf{T}} \Pi'$) iff, given an oracle $\square$ optimally solving Π', we can devise an algorithm optimally solving Π, in polynomial time if $\square$ is polynomial. ▮

The other two reductions, denoted by FT and DFT, respectively, have mainly the property of preserving membership in **FPTAS** and **DFPTAS**, respectively. Let Π and Π' be two **NP** maximization problems. Let $\square_{\alpha}^{\Pi'}$ be an oracle for Π' producing, for any $\alpha \in]0,1]$ and for any instance x' of Π', a feasible solution $\square_{\alpha}^{\Pi'}(x')$ of x' that is an $(1-\alpha)$-approximation for the standard ratio.

Definition 4. Π FT-reduces to Π' (denoted by $\Pi \leq_{\mathsf{FT}} \Pi'$) iff, for any $\varepsilon > 0$, there exists an algorithm $\mathtt{A}_{\varepsilon}(x, \square_{\alpha}^{\Pi'})$ such that:

- for any instance x of Π, $\mathtt{A}_{\varepsilon}$ returns a feasible solution which is a $(1-\varepsilon)$-standard approximation;
- if $\square_{\alpha}^{\Pi'}(x')$ works in time polynomial in both $|x'|$ and $1/\alpha$, then $\mathtt{A}_{\varepsilon}$ is polynomial in both $|x|$ and $1/\varepsilon$. ▮

For the case where at least one among Π and Π' is a minimization problem it suffices to replace $1-\epsilon$ or/and $1-\alpha$ by $1+\epsilon$ or/and $1+\alpha$, respectively.

Clearly, reduction of Definition 4 transforms a fully polynomial time approximation schema for Π' into a fully polynomial time approximation schema for Π. Reduction DFT, dealing with differential approximation, can be defined analogously.

Proposition 1. *$\leq_{\mathsf{FT}}$ (resp., $\leq_{\mathsf{DFT}}$) is reflexive, transitive, and preserves membership in **FPTAS** (resp., **DFPTAS**).*

We recall now the DPTAS-reduction by means of which, the existence of **DAPX**-complete problems has been proved in [1]. It will be useful in Section 6. Consider two **NPO** problems Π and Π'. Then, $\Pi \leq_{\mathsf{DPTAS}} \Pi'$ if there exist three functions f, g and c, computable in polynomial time, such that:

- $\forall x \in \mathcal{I}_{\Pi}$, $\forall \epsilon \in]0, 1[\cap \mathbb{Q}$, $f(x, \epsilon) \in \mathcal{I}_{\Pi'}$; f is possibly multi-valued;
- $\forall x \in \mathcal{I}_{\Pi}$, $\forall \epsilon \in]0, 1[\cap \mathbb{Q}$, $\forall y \in \mathrm{sol}_{\Pi'}(f(x, \epsilon))$, $g(x, y, \epsilon) \in \mathrm{sol}_{\Pi}(x)$;
- $c :]0, 1[\cap \mathbb{Q} \rightarrow]0, 1[\cap \mathbb{Q}$;
- $\forall x \in \mathcal{I}_{\Pi}$, $\forall \epsilon \in]0, 1[\cap \mathbb{Q}$, $\forall y \in \mathrm{sol}_{\Pi'}(f(x, \epsilon))$, $\delta_{\Pi'}(f(x, \epsilon), y) \geqslant 1 - c(\epsilon) \Rightarrow \delta_{\Pi}(x, g(x, y, \epsilon)) \geqslant 1 - \epsilon$; if f is multi-valued, i.e., $f = (f_1, \ldots, f_i)$, for some i polynomial in $|x|$, then, the former implication becomes: $\forall x \in \mathcal{I}_{\Pi}$, $\forall \epsilon \in]0, 1[\cap \mathbb{Q}$, $\forall y \in \mathrm{sol}_{\Pi'}((f_1, \ldots, f_i)(x, \epsilon))$, $\exists j \leqslant i$ such that $\delta_{\Pi'}(f_j(x, \epsilon), y) \geqslant 1 - c(\epsilon) \Rightarrow \delta_{\Pi}(x, g(x, y, \epsilon)) \geqslant 1 - \epsilon$. ▮

It can be easily shown that given two **NPO** problems Π and Π', if $\Pi \leq_{\mathsf{DPTAS}} \Pi'$ and $\Pi' \in$ **DPTAS**, then $\Pi \in$ **DPTAS**. One of the basic features of differential approximation ratio is that it is stable under affine transformations of the objective functions of the problems dealt. In this sense, problems for which the objective functions of the ones are affine transformations of the objective functions of the others, are approximate equivalent for the differential approximation paradigm (this is absolutely not the case for standard paradigm). The most notorious case of such problems is the pair MAX INDEPENDENT SET and MIN VERTEX COVER. Affine transformation is nothing else than a kind of reduction, denoted by AF, in what follows. Two problems Π and Π' are affine equivalent if $\Pi \leq_{\mathsf{AF}} \Pi'$ and $\Pi' \leq_{\mathsf{AF}} \Pi$. Obviously affine transformation is both an DFT- and a DPTAS-reduction.

We finally recall the F-reduction introduced in [5]. Consider two **NPO** problems Π and Π', Π F-reduces to Π' if and only if there exist three polynomially computable functions f, g and c such that:

- $\forall x \in \mathcal{I}_{\Pi}$, $f(x) \in \mathcal{I}_{\Pi'}$;
- $\forall I \in \mathcal{I}_{\Pi}$, $\forall y \in \mathrm{Sol}_{\Pi'}(f(x))$, $g(x, y) \in \mathrm{Sol}_{\Pi}(x)$;
- $c : \mathcal{I}_{\Pi} \times (]0, 1[\cap \mathbb{Q}) \rightarrow]0, 1[\cap \mathbb{Q}$; there exists a polynomial p such that $c(x, \epsilon) = 1/p(|x|, 1/\epsilon)$; moreover, $\forall x \in \mathcal{I}_{\Pi}$, $\forall \epsilon \in]0, 1[\cap \mathbb{Q}$, $\forall y \in \mathrm{Sol}_{\Pi'}(f(x))$, $\varepsilon(f(x), y) \leqslant c(x, \epsilon) \Rightarrow \varepsilon(x, g(x, y)) \leqslant \epsilon$, where, for an instance x of a problem in **NPO** and for a solution $y \in \mathrm{Sol}(x)$, $\varepsilon(x, y) = |\operatorname{opt}(x) - m(x, y)| / \operatorname{opt}(x)$.

Obviously, F-reduction preserves membership in **FPTAS**; furthermore it is a special case of FT-reduction since this latter one explicitely allows multiple calls to oracle $\square$ (this fact is not explicit in F-reduction; in other words, it is not clearly mentioned if f and g are allowed to be multivalued). Also, FT-reduction seems allowing more freedom in the way Π is transformed to Π'; for instance, in F-reduction, function g transforms an optimal solution for Π' into an optimal solution for Π, i.e., F-reduction preserves optimality; this is not the case for FT-reduction. This freedom will allow us in reducing non polynomially bounded **NPO** problems to **NPO-PB** ones. It seems so that the latter reduction is larger than the former one but this fact remains to be confirmed and such proof does not seem to be trivial and is not considered in this paper.

In what follows, given a class $\mathbf{C} \subseteq$ **NPO** and a reduction R, we denote by $\overline{\mathbf{C}}^{\mathsf{R}}$ the closure of **C** under R, i.e., the set of problems in **NPO** that R-reduce to some problem in **C**.

3 PTAS-completeness

We now study **PTAS**-completeness under FT-reduction. The following theorem introduces the main result of this section.

Theorem 1. *Let Π' be an* ***NP****-hard a problem of* ***NPO****. If $\Pi' \in$* ***NPO-PB****, then any* ***NPO*** *problem FT-reduces to Π'.*

The proof of Theorem 1 immediately follows from Lemmata 1 and 2. The first one introduces a property of Turing-reduction (Definition 3) for **NP**-hard problems. In the second one, we transform (under certain conditions) a Turing-reduction into a FT-reduction. Proofs of the two lemmata are given for maximization problems. The case of minimization is completely analogous.

Lemma 1. *If an* ***NPO*** *problem Π' is* ***NP****-hard, then any* ***NPO*** *problem Turing-reduces to Π'.*

Proof. Let Π be a **NPO** problem and q be a polynomial such that $|y| \leqslant q(|x|)$ for any instance x of Π and for any feasible solution y of x. Assume that encoding $n(y)$ of y is binary. Then $0 \leqslant n(y) \leqslant 2^{q(|x|)} - 1$. We consider the following problem $\hat{\Pi}$ (see [3]) which is the same as Π up to its objective function that is defined by $m_{\hat{\Pi}}(x,y) = 2^{q(|x|)+1} m_P(x,y) + n(y)$.

Clearly, if $m_{\hat{\Pi}}(x,y_1) \geqslant m_{\hat{\Pi}}(x,y_2)$, then $m_\Pi(x,y_1) \geqslant m_\Pi(x,y_2)$. So, if y is an optimal solution for the instance x of $\hat{\Pi}$, then it is also an optimal solution for the instance x of Π.

Remark now that for $\hat{\Pi}$, the evaluation problem $\hat{\Pi}_e$ and the constructive problem $\hat{\Pi}$ are equivalent. Indeed, given the value of an optimal solution y, one can determine $n(y)$ (hence y) by computing the remainder of the division of this value by $2^{q(|x|)+1}$.

Since Π' is **NP**-hard, we can solve the evaluation problem $\hat{\Pi}_e$ if we can solve the (constructive) problem Π'. Indeed,

- we can solve $\hat{\Pi}_e$ using an oracle solving the decision version $\hat{\Pi}_d$ of $\hat{\Pi}$, by dichotomy;
- $\hat{\Pi}_d$ reduces to the decision version Π'_d of Π' by a Karp-reduction (see [2, 9] for a formal definition of this reduction);
- finally, one can solve Π'_d using an oracle for the constructive problem Π'.

So, with a polynomial number of queries to an oracle solving Π', one can solve $\hat{\Pi}_e$, $\hat{\Pi}$ and the proof of the lemma is complete. ▮

We now show how, starting from a Turing-reduction (that only preserves optimality) between two **NPO** problems Π and Π' where Π' is polynomially bounded, one can obtain an FT-reduction transforming a fully polynomial time approximation schema for Π' into a fully polynomial time approximation schema for Π.

Lemma 2. *Let $\Pi' \in$ **NPO-PB**. Then, any **NPO** problem that is Turing-reducible to Π' is also FT-reducible to Π'.*

Proof. Let Π be an **NPO** problem and suppose that there exists a Turing-reduction between Π and Π'. Let $\square_\alpha^{\Pi'}$ be an oracle computing, for any instance x' of Π' and for any $\alpha > 0$, a feasible solution y' of x' such that $r(x', y') \geqslant 1 - \alpha$. Moreover, let p be a polynomial such that for any instance x' of Π' and for any feasible solution y' of x', $m(x', y') \leqslant p(|x'|)$.

Let x be an instance of Π. The Turing-reduction claimed gives an algorithm solving Π using an oracle for Π'. Consider now this algorithm where we use, for any query to the oracle with the instance x' of Π', the approximate oracle $\square_\alpha^{\Pi'}(x')$, with $\alpha = 1/(p(|x'|) + 1)$. This algorithm produces an optimal solution, since a solution y' being an $(1 - (1/(p(|x'|) + 1)))$-approximation for x' is an optimal one (recall that we deal with problems having integer-valued objective functions, cf., Definition 1). Really,

$$\begin{aligned}\frac{m_{\Pi'}(x', y')}{\operatorname{opt}_{\Pi'}(x')} \geqslant 1 - \frac{1}{p(|x'|) + 1} &\implies m_{\Pi'}(x', y') > \operatorname{opt}_{\Pi'}(x') - 1 \\ &\implies m_{\Pi'}(x', y') = \operatorname{opt}(x')\end{aligned}$$

It's easy to see that this algorithm is polynomial when $\square_\alpha^{\Pi'}(x')$ is polynomial in $|x'|$ and in $1/\alpha$.

Obviously, any exact algorithm for Π can be a posteriori seen as a fully polynomial time approximation schema; so, $\Pi \leq_{\mathsf{FT}} \Pi'$ and the proof of the lemma is now complete. ▮

From Theorem 1, one can immediately deduce the two following corollaries.

Corollary 1. $\overline{\textbf{\textit{PTAS}}}^{\mathsf{FT}} = \textbf{\textit{NPO}}$.

Corollary 2. *Any polynomially bounded problem in **PTAS** is **PTAS**-complete under FT-reduction.*

For instance, MAX PLANAR INDEPENDENT SET and MIN PLANAR VERTEX COVER are in **PTAS** ([4]). What has been discussed in this section concludes then the following result.

Theorem 2. MAX PLANAR INDEPENDENT SET *and* MIN PLANAR VERTEX COVER *are **PTAS**-complete under FT-reduction.*

Remark that the results of Theorem 2 cannot be trivially obtained using the F-reduction of [5].

4 DPTAS-completeness

We study in this section **DPTAS**-completeness under DFT-reduction. The results we shall obtain are analogous to the case of the **DPTAS**-completeness: we show that any **NPO-DPB NP**-hard problem in **DPTAS** is **DPTAS**-complete.

Theorem 3. *Let Π' be an **NPO-DPB NP**-hard problem. Then any problem in **NPO** is DFT-reducible to Π'.*

Theorem 3 is an immediate consequence of Lemma 1 and of the following lemma, differential counterpart of Lemma 2.

Lemma 3. *If $\Pi' \in$ **NPO-DPB**, then any **NPO** problem that is Turing-reducible to Π' is also DFT-reducible to Π'.*

Proof. Let Π be an **NPO** problem, and suppose that $\Pi \leq_{\mathsf{T}} \Pi'$. Let $\square_{\alpha}^{\Pi'}$ be an oracle computing, for any instance x' of Π' and for every $\alpha > 0$, a feasible solution y' such that $\delta(x', y') \geqslant (1 - \alpha)$. Let p be a polynomial such that for any instance x' of Π', $|\operatorname{opt}(x') - \omega(x')| \leqslant p(|x'|)$.

In the same way as in Lemma 2, we modify the algorithm of the Turing-reduction between Π and Π' using the approximate oracle $\square_{\alpha}^{\Pi}$ with $\alpha = 1/(p(|x'|) + 1)$. This algorithm computes, as in Lemma 2, an optimal solution and it is polynomial if the oracle is polynomial in $|x'|$ and in $1/\alpha$. This algorithm is obviously a differential fully polynomial time approximation schema, and hence, $\Pi \leq_{\mathsf{DFT}} \Pi'$. ∎

Corollary 3. $\overline{\textbf{\textit{DPTAS}}}^{\mathsf{DFT}} = \textbf{\textit{NPO}}$.

Corollary 4. *Any **NPO-DPB** problem in **DPTAS** is **DPTAS**-complete under DFT-reductions.*

The following concluding theorem deals with the existence of **DPTAS**-complete problems.

Theorem 4. *Problems* MAX PLANAR INDEPENDENT SET, MIN PLANAR VERTEX COVER *and* BIN PACKING *are **DPTAS**-complete under DFT-reduction.*

Proof. For the **DPTAS**-completenes of MAX PLANAR INDEPENDENT SET, just observe that for any instance G, $\omega(G) = 0$. So, standard and differential approximation ratios coincide for this problem; moreover, it is in both **NPO-PB** and **NPO-DPB**. Then, inclusion MAX PLANAR INDEPENDENT SET $\in$ **PTAS** suffices to conclude MAX PLANAR INDEPENDENT SET $\in$ **DPTAS** and, by Corollary 4, that it is **DPTAS**-complete.

MAX PLANAR INDEPENDENT SET and MIN PLANAR VERTEX COVER are affine equivalent; hence MAX PLANAR INDEPENDENT SET $\leq_{\mathsf{AF}}$ MIN PLANAR VERTEX COVER. Since AF-reduction is a particular kind of DFT-reduction, the **DPTAS**-completeness of MIN PLANAR VERTEX COVER is immediately concluded.

Finally, the **DPTAS**-completeness of BIN PACKING is concluded from the facts: (i) BIN PACKING $\in$ **DPTAS** ([7]) and (ii) BIN PACKING $\in$ **NPO-DPB** (since, for any instance L of size n, $\omega(L) = n$ and $\mathrm{opt}(L) > 0$. ∎

5 Looking for intermediate problems

FT-reduction is weaker than the F-reduction of [5] and, as we mentioned before, there exist **PTAS**-intermediate problems under this latter reduction. The question of existence of such problems is posed for our reduction too. In this section, we partially answer this question via the following theorem.

Theorem 5. *If there exists an **NPO**-intermediate problem for the Turing-reduction, then there exists a problem **PTAS**-intermediate for FT-reductions.*

Proof. Let Π be an **NPO** problem, intermediate for the Turing-reduction. Suppose that Π is a maximization problem (the minimization case is completely similar). Let p be a polynomial such that, for any instance x and any feasible solution y of x, $m(x,y) \leqslant 2^{q(|x|)}$. Consider the following maximization problem $\widetilde{\Pi}$ where:

- instances are the pairs (x,k) with x an instance of Π and k an integer in $\{0,\ldots 2^{q(|x|)}\}$;
- for an instance (x,k) of $\widetilde{\Pi}$, its feasible solutions are the feasible solutions of the instance x of Π;
- the objective function of $\widetilde{\Pi}$ is:

$$m_{\widetilde{\Pi}}((x,k),y) = \begin{cases} |(x,k)| & \text{if } v(x,y) \geqslant k \\ |(x,k)| - 1 & \text{otherwise} \end{cases}$$

We will now show the three following properties:

1. $\widetilde{\Pi} \in$ **PTAS**;
2. If $\widetilde{\Pi}$ were in **FPTAS**, then Π would be polynomial;
3. if $\widetilde{\Pi}$ were **PTAS**-complete, then Π would be **NPO**-complete under Turing-reductions.

If Properties 1, 2 and 3 hold, then since Π is supposed to be intermediate, one can conclude that $\widetilde{\Pi}$ is **PTAS**-intermediate, under FT.

Proof of Property 1. Remark that $\widetilde{\Pi}$ is clearly in **NPO-PB**. Consider $\varepsilon \in]0,1]$ and the algorithm $\mathtt{A}_\varepsilon$ which, on the instance (x,k) of $\widetilde{\Pi}$, solves exactly (x,k), if $|(x,k)| \leqslant 1/\varepsilon$; otherwise, it produces some solution. Algorithm $\mathtt{A}_\varepsilon$ is polynomial and guarantees standard approximation ratio, $1-\varepsilon$. Therefore, $\widetilde{\Pi}$ is in **PTAS**.

Proof of Property 2. Remark that $\Pi \leq_{\mathsf{T}} \widetilde{\Pi}$. Indeed, let x be an instance of Π. We can find an optimal solution of x solving $\log(2^{p(|x|)}) = p(|x|)$ instances (x,k) of $\widetilde{\Pi}$ (by dichotomy). Note that if $\widetilde{\Pi}$ were in **FPTAS**, it would be polynomial since the fully polynomial time approximation schema $\mathtt{A}_\varepsilon$ applied on instance (x,k) with $\varepsilon = 1/(|(x,k)|+1)$ is an exact and polynomial algorithm. The fact that $\Pi \leq_{\mathsf{T}} \widetilde{\Pi}$ would imply in this case that Π is polynomial.

Proof of Property 3. Assume that $\widetilde{\Pi}$ is **PTAS**-complete (under FT-reductions). Then, MAX PLANAR INDEPENDENT SET FT-reduces to $\widetilde{\Pi}$. Let $\Box$ be an oracle solving Π. Then, we immediately obtain an exact algorithm for $\widetilde{\Pi}$, polynomial if $\Box$ is so. Clearly, this algorithm can be considered as a fully polynomial time approximation schema for $\widetilde{\Pi}$. Reduction MAX PLANAR INDEPENDENT SET $\leq_{\mathsf{FT}}$ $\widetilde{\Pi}$ provides a fully polynomial time approximation schema for MAX PLANAR INDEPENDENT SET and, since it is in **NPO-PB**, we get an exact (and polynomial if $\Box$ is so) algorithm for it. In other words, if $\widetilde{\Pi}$ is **PTAS**-complete, then MAX PLANAR INDEPENDENT SET $\leq_{\mathsf{T}}$ Π. To conclude, MAX PLANAR INDEPENDENT SET is **NPO**-complete under Turing-reduction, since it is **NP**-hard (cf., Lemma 1). Therefore, if $\widetilde{\Pi}$ were **PTAS**-complete, Π would be **NPO**-complete (under Turing-reductions). The proof of Property 3 and of the theorem are now completed. ▮

We now state an analogous result about the existence of **DPTAS**-intermediate problems (under DFT-reductions).

Theorem 6. *If there exists an **NPO**-intermediate problem under Turing-reductions, then there exists a problem **DPTAS**-intermediate, under DFT-reductions.*

Proof. The proof is analogous to one of Theorem 5, up to modification of definition of $\widetilde{\Pi}$. Indeed, if we don't modify it, then $\widetilde{\Pi}$ is not in **DPTAS**, because the value of the worst solution of an instance (x,k) is $|(x,k)|-1$. We only have to change this definition in order to have $\omega((x,k)) = 0$ for any instance (x,k). For instance, we can define $\widetilde{\Pi}$ as follows:

- instances of $\widetilde{\Pi}$ are, as previously, the pairs (x,k) where x is an instance of Π and k is an integer between 0 and $2^{q(|x|)}$;
- for an instance (x,k) of $\widetilde{\Pi}$, its feasible solutions are the feasible solutions of the instance x of Π, plus a solution y_x^0;
- the objective function of $\widetilde{\Pi}$ is:

$$m_{\widetilde{\Pi}}((x,k),y) = \begin{cases} 0 & \text{if } y = y_x^0 \\ |(x,k)| & \text{if } v(x,y) \geqslant k \\ |(x,k)|-1 & \text{otherwise} \end{cases}$$

Then, the result claimed is get in exactly the same way as in the proof of Theorem 5. ▮

6 A new DAPX-complete problem

All **DAPX**-complete problems given in [1] are also **APX**-complete under **E**-reduction ([11]), a generalization of the L-reduction of [13]. An interesting question is if there exist **DAPX**-complete problems that are not also **APX**-complete for some standard approximation-preserving reduction. In this section, we positively answer this question by the following theorem.

Theorem 7. MIN COLORING *is **DAPX**-complete under DPTAS-reductions.*

Proof. Consider problem MAX UNUSED COLORS and remark that standard ratio for it coincides with differential ratio of MIN COLORING. In fact, these problems are affine equivalent; so, a posteriori

$$\text{MAX UNUSED COLORS} \leq_{\mathsf{AF}} \text{MIN COLORING} \tag{1}$$

MAX UNUSED COLORS has been proved **MAX-SNP**-hard under L-reduction ([10]). Moreover, as it is shown in [11], $\overline{\textbf{MAX-SNP}}^{\mathsf{E}} = \textbf{APX-PB}$ (the set **NPO-PB** $\cap$ **APX**). Since MAX INDEPENDENT SET-$B \in$ **APX-PB**, MAX INDEPENDENT SET-$B \leq_{\mathsf{E}}$ MAX UNUSED COLORS. On the other hand, E-reduction being a particular kind of PTAS-reduction, MAX INDEPENDENT SET-$B \leq_{\mathsf{PTAS}}$ MAX UNUSED COLORS. Note now that this PTAS-reduction is simultaneously a DPTAS-reduction between MAX INDEPENDENT SET-B and MIN COLORING. In fact, standard and differential approximation ratios for MAX INDEPENDENT SET-B, on the one hand, standard and differential approximation ratios for MAX UNUSED COLORS and differential ratio of MIN COLORING, on the other hand, coincide. So,

$$\text{MAX INDEPENDENT SET-}B \leq_{\mathsf{DPTAS}} \text{MAX UNUSED COLORS} \tag{2}$$

Reductions (1) and (2), together with the fact that the composition DPTAS $\circ$ AF is obviously a DPTAS-reduction, establish immediately the **DAPX**-completeness of MIN COLORING and the proof of the theorem is now complete. ▮

As we have already mentioned, MIN COLORING is, until now, the only problem known to be **DAPX**-complete but not **APX**-complete. In fact, in standard approximation, it belongs to the class **Poly-APX** (of problems for which the best standard ratio known is a polynomial on the size of their instances) and is inapproximable, in a graph of order n, within $n^{1-\epsilon}$, $\forall\epsilon > 0$, unless **NP** coincides with the class of problems that could be optimally solved by slightly super-polynomial algorithms ([8]).

7 Conclusion

We have defined suitable reductions and obtained natural complete problems for classes **PTAS** and **DPTAS** of problems admitting a polynomial time approximation schema, in standard and differential approximation. This work extends the one in [1]; both aim in studying a structure of differential approximation classes.

However, the fact that problems proved complete here do not admit a standard or differential fully polynomial time approximation schema is already known. It would be interesting to use reductions proposed in order to get new inapproximability results. This is in fact a major computational impact of structuring approximation classes. Another interesting open question concerns relationships between F and FT-reductions; for example, is the latter strictly weaker than the former? Finally, the existence of natural **PTAS**-, or **DPTAS**-intermediate problems (as BIN PACKING for **APX** under AP-reduction) for F-, FT and DFT-reductions remains open.

References

[1] G. Ausiello, C. Bazgan, M. Demange, and V. Th. Paschos. Completeness in differential approximation. In *Mathematical Foundations of Computer Science*. Springer-Verlag, 2003.

[2] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and approximation. Combinatorial optimization problems and their approximability properties*. Springer, 1999.

[3] G. Ausiello, P. Crescenzi, and M. Protasi. Approximate solution of NP optimization problems. *Theoretical Computer Science*, 150(1):1–55, 1995.

[4] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the Association for Computing Machinery*, 41(1):153–180, 1994.

[5] P. Crescenzi and A. Panconesi. Completeness in approximation classes. *Information and Computation*, 93(2):241–262, 1991.

[6] P. Crescenzi and L. Trevisan. On approximation scheme preserving reducibility and its applications. *Theory of Computing Systems*, 33(1):1–16, 2000.

[7] M. Demange, J. Monnot, and V. Th. Paschos. Bridging gap between standard and differential polynomial approximation : the case of bin-packing. *Applied Mathematics Letters*, 12:127–133, 1999.

[8] U. Feige and J. Kilian. Zero knowledge and the chromatic number. In *Proceedings of the Conference of Computational Complexity*, pages 278–287, 1996.

[9] M. R. Garey and D. S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. Freeman, 1979.

[10] M. M. Halldorsson. Approximating discrete collections via local improvements. In *Proceedings of the Symposium on Discrete Algorithms*, pages 160–169, 1995.

[11] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani. On syntactic versus computational views of approximability. *SIAM Journal on Computing*, 28(1):164–191, 1998.

[12] R. E. Ladner. On the structure of polynomial time reducibility. *Journal of the Association for Computing Machinery*, 22:155–171, 1975.

[13] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[14] C. H. Papadimitriou and M. Yannakakis. The traveling salesman problem with distances one and two. *Mathematics of Operations Research*, 18:1–11, 1993.

A A list of NPO problems

We present the list of **NPO** problems mentioned and/or discussed in the paper, together with a characterization of their worst-value solutions. For most of these problems, comments about their approximability in standard approximation can be found in [2].

Maximum variable-weighted satisfiability.
Given a boolean formula φ with non-negative integer weights $w(x)$ on any variable x appearing in φ, maximum variable-weighted satisfiability consists of computing a truth assignment to the variables of φ that both satisfies φ and maximizes the sum of the weights of the variables set to 1. We consider that the assignment setting all the variables to 0, even if it does not satisfy φ, is feasible and represents the worst-value solution for the problem. Maximum linear variable-weighted satisfiability-B denotes the version of Maximum linear variable-weighted satisfiability where the variable-weights are polynomially bounded and their sum lies in the interval $[B, (n/(n-1))B]$. For this problem, it is assumed that the assignment setting all variables to 0 is feasible and that its value is B. Obviously, this assignment represents the worst feasible value.

Maximum independent set (MAX INDEPENDENT SET).
Given a graph $G(V, E)$, an *independent set* is a subset $V' \subseteq V$ such that whenever $\{v_i, v_j\} \subseteq V'$, $v_iv_j \notin E$, and MAX INDEPENDENT SET consists in finding an independent set of maximum size. By MAX INDEPENDENT SET-B, we denote MAX INDEPENDENT SET in bounded-degree graphs. Finally, by MAX PLANAR INDEPENDENT SET, we denote MAX INDEPENDENT SET in planar graphs. Worst-value solution: the empty set.

Minimum coloring (MIN COLORING) and maximum color saving (MAX UNUSED COLORS).
Given a graph $G(V, E)$, we wish to color V with as few colors as possible so that no two adjacent vertices receive the same color. Worst-value solution: V. MAX UNUSED COLORS is the problem consisting, given a a graph $G(V, E)$ and a set of $|V|$ colors, of coloring G using colors from the set given, in such a way that the number of unused colors is maximized. Clearly, since the initial set is feasible, worst solution for this problem is exactly this set. It can be immediately seen that the complement of a legal coloring with respect to the vertex-set V is a feasible solution for MAX UNUSED COLORS; on the other hand, the complement of a feasible solution with respect to the set of initial colors, is a feasible solution for MIN COLORING. In other words, MIN COLORING and MAX UNUSED COLORS are affine equivalent.

Minimum vertex-covering (MIN VERTEX COVER).
Given a graph $G(V, E)$, a *vertex cover* is a subset $V' \subseteq V$ such that, $\forall uv \in E$, either $u \in V'$, or $v \in V'$, and MIN VERTEX COVER consists of determining a minimum-size vertex cover. By MIN PLANAR VERTEX COVER, we denote MIN VERTEX COVER in planar graphs. Worst-value solution: V.

Bin packing (BIN PACKING).
Given a finite set $L = \{x_1, \ldots, x_n\}$ of n rational numbers and an unbounded number of bins, each bin having a capacity equal to 1; we wish to arrange all these numbers in the least possible bins in such a way that the sum of the numbers in each bin does not violate its capacity. Worst solution: L.

Minimum traveling salesman problem (MIN TSP).
Given a complete graph on n vertices, denoted by K_n, with positive distances on its edges, MIN TSP consists of minimizing the cost of a Hamiltonian cycle (an ordering $\langle v_1, v_2, \ldots, v_n\rangle$ of V such that $v_n v_1 \in E$ and, for $1 \leqslant i < n$, $v_i v_{i+1} \in E$), the cost of such a cycle being the sum of the distances of its edges. We denote by MIN METRIC TSP the version of MIN TSP where edge distances satisfy triangle inequalities. Worst-value solution: the total distance of the longest Hamiltonian cycle.

Un mécanisme de négociation multicritère pour le commerce électronique

Marie-Jo Bellosta [*], Imène Brigui [*], Sylvie Kornman [*], Suzanne Pinson [*], Daniel Vanderpooten [*]

Résumé

Dans cet article nous présentons un mécanisme de négociation multicritère pour le commerce électronique fondé sur un modèle multicritère utilisant des points de référence. Selon ce modèle, l'acheteur doit spécifier un point d'aspiration qui exprime les valeurs souhaitées sur chaque attribut décrivant le produit à acheter et un point d'exigence qui représente les valeurs minimales acceptables sur chaque critère. Le mécanisme de négociation utilise un protocole d'enchères anglaises inversées et conduit la négociation vers le point d'aspiration de l'acheteur en assurant un contrôle direct sur le processus d'échanges.

Mots-clefs : Négociation, enchères multi-attributs, système multiagents, commerce électronique

Abstract

In this paper we present a multi-attribute negotiation mechanism for electronic commerce which is based on a multicriteria model using reference points. According to the model, the buyer must specify an aspiration point that expresses his desired values on the attributes and a reservation point that represents the minimal values required. The negotiation mechanism uses an English reverse auction protocol and leads the negotiation to the buyer's aspiration point by providing a direct control on the bidding process.

Key words : Negotiation, multi-attribute auctions, multi-agent systems, electronic commerce

[*] LAMSADE, Université Paris-Dauphine, 75775 Paris cedex 16, France. {bellosta, brigui, kornman, pinson, vdp}@lamsade.dauphine.fr

1 Introduction

Les applications de commerce électronique de dernière génération suivent généralement le modèle Consumer Buying Behaviour model (CBB) [7]. Le modèle CBB comporte six phases: identification des besoins, sélection des produits, recherche des meilleures offres, négociation, achat et livraison du produit. Les acteurs humains consommateurs et vendeurs sont représentés par des agents logiciels. Un consommateur est représenté par un agent acheteur qui connaît les produits cherchés, ainsi que les préférences du consommateur. Un vendeur est représenté par un agent vendeur qui connaît l'état des stocks et les conditions de vente que le vendeur est prêt à accorder. La phase de négociation porte sur l'amélioration des conditions de vente. L'accord final entre agents résulte d'une suite d'échanges régis par un protocole de négociation. Les protocoles d'enchères offrent les mécanismes les plus compétitifs pour la négociation [8, 17]. Un mécanisme d'enchères alloue des ressources aux acheteurs et aux vendeurs en se basant sur des règles prédéfinies. Ces règles définissent le processus d'échange de propositions, la détermination du gagnant et l'accord final. Les protocoles d'enchères mis en oeuvre sont les enchères hollandaises, les enchères à enveloppes scellées, les enchères Vickrey [16] et les enchères anglaises. Face aux enchères portant uniquement sur le prix qui sont largement dominantes [5, 9, 12], d'autres types d'enchères ont été définis et étudiés tels que les enchères à multiples exemplaires [3], les enchères combinées [6, 13] et les enchères multi-attributs [1, 3, 5, 10, 11]. Les enchères à multiples exemplaires portent sur un ensemble d'articles identiques. Les enchères combinées sont des enchères à multiples exemplaires où les propositions peuvent s'effectuer sur une partie de l'ensemble d'articles. Les enchères multi-attributs portent sur plusieurs caractéristiques d'un produit incluant non seulement son prix mais aussi sa qualité, les conditions de livraisons, de maintenance, etc. Les acheteurs définissent leurs préférences sur l'article recherché et les vendeurs sont en concurrence sur tous les attributs spécifiés par l'acheteur. Dans cet article, nous prenons en compte les enchères multi-attributs.

L'automatisation d'enchères multi-attributs s'appuie sur plusieurs composants clefs:

1. un modèle de préférence pour l'expression des préférences de l'acheteur;
2. une méthode d'agrégation multicritère pour la sélection de la meilleure offre par l'agent acheteur;
3. un module de décision de l'agent acheteur pour la formulation des contre propositions.

Les méthodes d'agrégation multicritères se basent généralement sur le modèle de la somme pondérée. Selon ce modèle, chaque critère se voit assigner un poids rendant compte de son importance. L'évaluation d'une proposition se fait en sommant les valeurs pondérées relatives à chaque critère. La meilleure proposition correspond à l'évaluation la plus élevée. Toutefois, ce modèle présente quelques défauts [15] :

1. les poids associés aux attributs sont difficiles à définir et à interpréter, d'autant plus que de petites variations de ces poids peuvent changer radicalement le choix de la meilleure proposition;

2. le modèle de la somme pondérée est totalement compensatoire. Ainsi, une proposition avec un très bas score sur un critère important et des scores élevés sur d'autres critères à moindre importance peut être préférée à une autre proposition ayant de bons scores sur l'ensemble des critères.

3. une proposition non-dominée peut ne jamais être la meilleure. Ceci est un inconvénient sévère, car certaines solutions non dominées sont systématiquement rejetées pour d'uniques raisons techniques.

Dans cet article, nous proposons un mécanisme de négociation basé sur un modèle à points de référence pour répondre aux insuffisances du modèle de la somme pondérée. Le mécanisme proposé adapte le protocole d'enchères anglaises inversées au modèle multicritère à points de référence. A chaque étape de la négociation, l'agent acheteur spécifie la valeur minimale acceptable sur chaque critère d'une proposition. Ce mécanisme permet à l'agent acheteur un contrôle plus direct sur le processus de négociation.

La suite de cet article est structurée en six parties. La section 2 présente un état de l'art ciblant les enchères multi-attributs dans le commerce électronique. La section 3 introduit le modèle de préférence choisi. La section 4 détaille le mécanisme d'enchères adopté. La section 5 présente les tests réalisés et compare les résultats obtenus avec le modèle de la somme pondérée. La section 6 termine par une conclusion et des perspectives.

2 Etat de l'art

La plupart des recherches menées pour automatiser les enchères multi-attributs se basent sur le modèle de la somme pondérée [11, 4, 3, 10]. Les protocoles d'enchères classiques ont été étendus en remplaçant l'attribut prix par l'évaluation multi-attribut d'une proposition.

Oliveira et al. proposent un protocole peer-to-peer d'enchères anglaises inversées multi-attributs, basé sur la somme pondérée. La négociation est distribuée : l'agent acheteur négocie directement avec chaque vendeur. Les enchères comportent plusieurs étapes d'une durée prédéfinie. La négociation est lancée par l'agent acheteur qui envoie aux agents vendeurs concernés ses préférences sur le produit recherché ainsi que l'évaluation minimale de départ pour une proposition. Les vendeurs évaluent la demande, envoient une proposition acceptable ou se retirent de la négociation. Quand l'acheteur a reçu toutes les offres ou le délai prédéfini a été atteint, les offres sont évaluées et la meilleure sélectionnée. Son évaluation sert de base à l'étape suivante sous la forme d'une contre-proposition envoyée par l'agent acheteur aux vendeurs restant en compétition. Les enchères continuent jusqu`à ce que tous les vendeurs sauf un aient abandonné et se terminent sur un contrat avec le vendeur gagnant (le dernier restant). Ce protocole a été défini pour prendre en compte l'aspect multicritère des enchères. Nous l'avons adapté dans le cadre du modèle multicritère à points de référence, ce qui nous a permis de comparer l'utilisation de notre modèle et celui de la somme pondérée.

Bichler et al. utilisent un protocole d'enchères multi-attributs pour une place de marché. Ils introduisent la notion de monnaie virtuelle, qui représente l'évaluation d'une proposition suivant son prix et la valeur des autres attributs. Les enchères utilisent un agent médiateur et débutent lorsque les agents vendeurs déclarent leurs capacités. Dans une étape ultérieure, l'acheteur pourvoit toutes les informations relatives à l'article souhaité et à ses préférences. Toutes ces informations sont regroupées dans un message d'appel à proposition envoyé à l'agent médiateur qui se charge d'en informer les différents vendeurs engagés dans la négociation et collecte en conséquence leurs propositions. Du côté des vendeurs, le système fournit un outil d'aide à la formulation des propositions leur permettant de demander des informations (à caractère anonyme) concernant les autres propositions. Une fois les propositions faites et l'enchère close, l'agent médiateur détermine la meilleure proposition toujours sur la base de la somme pondérée et établit le contrat. L'inconvénient majeur du modèle de la somme pondérée est son caractère totalement compensatoire, déjà évoqué dans l'introduction. Bichler [3] note que l'interprétation des poids associés à chaque attribut n'est pas claire et intuitive. Le modèle que nous proposons est basé sur des éléments faciles à interpréter (point d'exigence et point d'aspiration) et simplifie ainsi la phase d'élicitation des préférences.

Basé sur un modèle de préférences dédié à la vente de billets d'avion, le système SARDINE [10] présente un marché aux enchères décrit par plusieurs attributs statiques. Les préférences portent sur plusieurs critères tels que le prix, la date et l'heure du vol, auxquels est associé un degré de flexibilité pouvant prendre trois valeurs prédéfinies : *très-flexible*, *moyennement-flexible* et *peu-flexible* qui permet de dégager le poids ($weight$) et l'intervalle d'acceptabilité ($range$) de chaque critère. L'évaluation des propositions se fait suivant la formule suivante :

$$\text{dist} = \sum_i \text{weight}_i \left(\frac{\text{preferred}_i - \text{actual}_i}{\text{range}_i} \right)$$

- $Preferred_i$: la valeur préférée sur le ième critère.

- $Actual_i$: la valeur du ième critère dans la proposition courante.

La similarité à noter entre ce modèle et celui que nous proposons est la définition d'un point de référence exprimant les valeurs souhaitées sur chaque critère. Contrairement au modèle à points de référence, le modèle de SARDINE considère comme équivalentes les valeurs *(preferred-x)* et *(preferred+x)* d'un même attribut. Ainsi, si l'heure de départ souhaitée est 8h du matin, les heures de départ 11h et 5h du matin seront considérées comme équivalentes, ce qui n'est pas nécessairement le cas.

3 Modèle multicritère

Dans ce paragraphe, nous présentons les concepts fondamentaux du modèle multicritère qui permettent d'identifier les préférences de l'acheteur, ainsi que la méthode multicritère d'évaluation et de sélection des propositions.

3.1 Modèle de préférences

Nous donnons quelques notations et définitions préliminaires:

- n, le nombre de vendeurs.
- p, le nombre d'attributs.
- $D = D_1 \times \ldots \times D_p$, l'espace de décision où D_j désigne le domaine de valeurs du critère j.
- $x = (x_1, \ldots, x_p) \in D$ la proposition du $vendeur_i$.
- $C = C_1 \times \ldots \times C_p$ l'espace des critères.
- v_j, la fonction de valorisation définie de D_j dans $C_j = [0{,}100]$ correspondant à l'attribut j.
- $b_i = (b_{i1}, \ldots, b_{ip}) \in C$ avec $b_{ij} = v_j(x_{ij})$, la proposition valorisée du vendeur i sur le critère j.

Nous rappelons, par ailleurs, les concepts suivants de l'aide à la décision [14]:

- Δ, la relation de dominance telle que :

$$b\Delta b^{'} \Leftrightarrow \forall j \in \{1, \ldots, p\} b_j \geq b_j^{'} \text{ et } \exists l \in \{1, \ldots, p\} : b_l > b_l^{'}.$$

- b est non dominée ssi il n'existe pas $b^{'}$ telle que $b^{'}\Delta b$.
- b est efficace ssi b is non-dominée. L'ensemble de toutes les proposition efficaces est noté E.

- $ideal = (ideal_1, \ldots, ideal_p)$ où $ideal_j = \max_{b \in E}(b_j)$, *le point idéal* formé des scores optimaux sur tous les critères recueillis séparément.
- $antiIdeal$ = $(antiIdeal_1, \ldots, antiIdeal_p)$ où $antiIdeal_j = \min_{b \in E}(b_j)$, le point antiIdeal formé des scores minimaux sur tous les critères recueillis séparément.

Le modèle multicritère utilisé se base sur les points de référence suivants :

- $a = (a_1, \ldots, a_p)$: le point d'aspiration où a_j représente le score souhaité par l'acheteur sur le critère j. L'acheteur fournit ses aspirations en valeurs effectives et le système les traduit en scores pour former les niveaux d'aspiration.
- $e = (e_1, \ldots, e_p)$: le point d'exigence où e_j représente la valeur minimale exigée par l'acheteur sur le critère j. L'acheteur fournit ses exigences en valeurs effectives et le système les traduit en scores pour former les niveaux d'exigence.

3.2 Méthode multicritère

La méthode multicritère utilise la définition d'une déviation au point d'aspiration qui mesure l'écart maximal entre les valeurs des critères. Soit le point d'aspiration a et une proposition b, la déviation de b à a est définie par la relation:

$$\text{deviation}(a,b) = \max_{j=1,\ldots,p}\{\lambda_j(a_j - b_j)\} \text{ où } \quad \lambda_j = 1/(\text{ideal}_j - \text{antiIdeal}_j). \tag{1}$$

La déviation retenue est la norme pondérée de Tchebychev. Cette déviation est calculée pour chaque critère en évaluant l'écart entre le niveau d'aspiration et le score de la proposition pondéré par l'écart entre le meilleur et le pire score sur ce critère. Cette pondération permet de ramener à une même échelle les écarts absolus sur les différents critères: un écart absolu de 1 sur deux critères différents peut en effet représenter des écarts d'importance très différente. On retient finalement l'écart maximal obtenu sur l'ensemble des critères. Cet écart est positif si le point d'aspiration a n'est pas dominé par la proposition b et négatif sinon.

Proposition 1 La relation de préférence ϕ définie sur E par :

$$b_m \phi b_i \Leftrightarrow deviation(a,b_m) < deviation(a,b_i)$$

est une relation d'ordre total sur E . L'ensemble des propositions reçues lors d'une enchère est totalement ordonné par cette relation. Ainsi, à chaque étape d'une enchère, la meilleure proposition b^* est celle qui minimise la déviation au point d'aspiration :

$$B^* = \{b \in B : \arg\min_{b \in B}\{\text{deviation}(a,b)\}\} \tag{2}$$

4 Protocole et mécanisme d'enchères

La mise en oeuvre d'enchères automatiques suppose la définition d'un protocole de communication entre agents et d'un mécanisme de relance des enchères. Le protocole fixe les possibilités d'initier une négociation, de répondre à un message et d'utiliser des séquences d'actions au sein du processus d'enchères [7, 8]. Le mécanisme de relance détermine à chaque étape quelles contraintes les nouvelles propositions doivent respecter.

4.1 Primitives et sémantique

Le protocole considéré définit une adaptation à l'aspect multicritère du protocole des enchères anglaises inversées [11]. Le tableau 1 regroupe les actes primitifs de communication du protocole ainsi que la sémantique qui leur est associée.

Le diagramme de la Figure 1 définit le modèle comportemental de l'agent acheteur. La négociation débute quand l'agent acheteur envoie un appel d'offre à tous les agents vendeurs potentiellement intéressés ($EtatInitial$ vers $Etat_1$). L'appel d'offre définit le produit recherché, les préférences concernant le produit et la valeur minimale requise pour une proposition. L'agent acheteur attend alors les propositions des vendeurs ($Etat_1$). A la fin de l'étape, l'agent acheteur évalue les propositions ($Etat_1$ à $Etat_2$). Trois situations peuvent alors se présenter :

1. toutes les réponses sont des refus et la négociation se termine sur un échec ($Etat_2$ à $Echec$) ;
2. au moins deux réponses contiennent une proposition pour le produit recherché. La meilleure proposition est alors sélectionnée, le vendeur correspondant mis en attente et la contre-proposition calculée et envoyée aux vendeurs restants ($Etat_2$ à $Etat_1$);
3. une seule réponse comporte une proposition, l'agent acheteur l'accepte, envoie un message d'acceptation et attend la validation du vendeur ($Etat_2$ à *Succès*). La négociation se termine sur un succès.

4.2 Définition des contre-propositions

La définition des contre-propositions assure la progression des propositions vers le point d'aspiration de l'acheteur et l'efficacité, au sens de Pareto, du mécanisme d'enchères proposé. Elle est basée sur la règle du *beat-the-quote (BQ)* introduite dans [17]. Selon le

Primitives	Sémantique	Contexte
CallForPropose(a,g,Preferences)	a lance les enchères avec le groupe de vendeurs g en donnant ses préférences sur le produit.	a suppose que les vendeurs peuvent fournir le produit désiré.
propose (v, a, bid)	v envoie une proposition à a	En réponse à un message $callForPropose$ ou à un $requestForPropose$
requestForPropose(a, g, counterproposal)	a demande au groupe de vendeurs restants d'améliorer leurs offres	En réponse aux messages $propose$
Accept (a, v)	a accepte la dernière proposition envoyée par v	En réponse à un message $propose$ et annonçant la fin des enchères
Reject (a, v)	a élimine v des enchères	En réponse à un message $propose$ et annonçant la fin des enchères
abort (v)	v abandonne les enchères	En réponse à un message $callForPropose$ ou à un message $requestForPropose$

TAB. 1 – *Primitives de dialogue*

principe des enchères anglaises, la règle du *BQ* impose que toute nouvelle proposition soit meilleure que la meilleure proposition reçue jusqu'alors. Lorsque l'évaluation des propositions se réduit à une fonction d'agrégation réelle, cette règle est mise en oeuvre en introduisant un incrément ε qui représente l'amélioration demandée à chaque étape.

Proposition 2 Une condition suffisante pour que le mécanisme d'enchère réponde à la règle du *BQ* est qu'à l'étape $t+1$ de l'enchère toute proposition b^{t+1} satisfasse:

$$\forall j \in \{1,\ldots,p\} \qquad b_j^{t+1} \geq a_j - (d^t - \varepsilon)/\lambda_j$$

où a_j désigne le niveau d'aspiration sur le critère j, d_t la déviation minimale au point d'aspiration à l'étape t et ε un décrément spécifié à l'avance. La relation (3) se déduit de la relation suivante imposée par la règle du *BQ* et de la définition (1). Cette règle demande que toute proposition b^{t+1} reçue à l'étape $t+1$ respecte (4)

deviation$(a,b^{t+1}) <$ deviation$(a,best^t) = d^t$

Définition du point d'éxigence Une condition suffisante pour (4) peut être exprimée par un point d'exigence e^{t+1} actif à l'étape $t+1$. Le point d'exigence de l'étape $t+1$ est défini par la relation (5)

$$\forall j \in \{1,\ldots,p\} \qquad e_j^{t+1} = \max\{a_j - (d^t - \varepsilon)/\lambda_j; e_j^1\}$$

La définition du point d'exigence à l'étape $t+1$ est se fait en reportant la déviation de la meilleure proposition d^t sur l'ensemble des critères en considérant le décrément ε opéré sur sur cette déviation. Afin de respecter le point d'exigence défini au début des enchères et d'interdire sa dégradation, le niveau d'exigence à toute étape des enchères sur un critère donné doit être supérieur au niveau spécifié à la première itération.

A l'étape $t+1$, l'agent acheteur envoie le point d'exigence défini par (5) comme contre-proposition à tous les vendeurs appelés à améliorer leurs propositions. La définition des contre-propositions sous forme de points d'exigence assure la progression des propositions vers le point d'aspiration. De plus, elle permet à l'agent acheteur de garder privé son point d'aspiration et son modèle d'agrégation.

La figure 2 illustre la détermination du point d'exigence à l'étape suivante en fonction d'une meilleure proposition à l'étape t et du décrément ε. Parmi les propositions respectant le point d'exigence e_t, la meilleure est notée $meilleure_t$. En considérant la déviation entre cette proposition et le point d'aspiration, le point d'exigence de l'étape *t+1* est défini en reportant cette déviation - ε (réalisée dans cet exemple sur le critère le plus pénalisant c_2) sur les critères c_1 et c_2.

4.3 Propriétés

Le mécanisme d'enchère ainsi défini présente plusieurs propriétés intéressantes [2]. Tout d'abord, la suite des points d'exigence $e^t(t \in 1,\ldots,derniere)$ est une suite croissante pour la relation d'ordre ϕ définie dans la section 3. Cette propriété résulte de l'inégalité suivante:

$\forall t \in \{1,\ldots,\text{derniere} - 1\}$ $deviation(a,e^{t+1}) < deviation(a,e^t)$

Cette propriété assure la progression des propositions vers le point d'aspiration, étape après étape.

Par ailleurs, la suite e^t $(t \in 1,..,derniere)$ est également croissante pour la relation de dominance :

$$\text{best}^{t+1}\Delta\text{best}^t \qquad (3)$$

En outre, la définition d'une contre-proposition assure deux propriétés nécessaires à une enchère:

- la dominance de proposition ou $bid - dominance$ exige qu'un vendeur offre toujours une proposition meilleure que sa dernière envoyée ;
- l'efficacité suppose que le vendeur avec la meilleure proposition gagne.

En effet, dans les enchères anglaises, deux cas peuvent se produire : soit le vendeur avec la meilleure proposition gagne avec une proposition juste au-dessus de la deuxième meilleure, soit le deuxième vendeur gagne avec une proposition juste en-dessous de la meilleure. Ce deuxième cas arrive quand l'incrément utilisé par l'agent acheteur est supérieur à la différence entre la meilleure proposition et la deuxième meilleure. Dans ce sens, le mécanisme présenté assure l'efficacité des enchères. En résumé, le mécanisme d'enchères que nous avons défini assure les règles de base des enchères anglaises et garantit une évolution des enchères vers le point d'aspiration défini par l'acheteur.

4.4 Algorithme

L'algorithme d'enchères étend l'algorithme des enchères anglaises inversées [11] en considérant le modèle des points de référence. Il se décompose en quatre étapes décrites ci-dessous. L'acheteur lance l'enchère avec le nom du produit cherché et la liste des vendeurs qui fournissent ce produit.

Collecte des informations. L'agent acheteur collecte les préférences de l'acheteur (les fonctions de valorisation des critères, les valeurs effectives pour les points d'aspiration et d'exigence), la durée maximale de l'enchère et l'incrément ε. Appel d'offre. L'agent acheteur calcule les points de référence en utilisant les fonctions de valorisation et la durée maximale d'une étape. Il envoie un appel d'offre (performative $callForPropose$) composé des fonctions de valorisation, du point d'exigence initial, du temps de fin de l'enchère, et de la durée maximale d'une étape.

définition des Lambdas. L'agent acheteur reçoit les premières réponses (messages $propose$ et/ou $abort$). Il définit les valeurs λ_j avec $j \in 1, \ldots, p$, qui sont utilisées durant toute l'enchère.

Boucle de l'enchère. L'agent acheteur répète les opérations suivantes jusqu'à la fin de l'enchère, i.e., l'ensemble des vendeurs en compétition est vide ou le temps de fin de négociation est atteint :

1. sélectionne la meilleure proposition comme proposition de référence pour l'étape suivante et met en attente le vendeur correspondant,
2. définit le nouveau point d'exigence,
3. envoie une nouvelle demande aux agents vendeurs (performative *requestForProposal*) hormis le vendeur en attente,
4. attend et collecte les propositions des vendeurs.

Fin de l'enchère. Les enchères échouent s'il n'y a pas de proposition, sinon les enchères se terminent avec succès. S'il ne reste qu'une seule proposition, celle-ci est gagnante et l'acheteur envoie un message d'acceptation à l'agent associé (performative $accept$). Dans le cas contraire, la durée de l'enchère est atteinte et plusieurs propositions restent en compétition. L'agent acheteur envoie un message d'acceptation à l'agent associé à la meilleure proposition et un message de rejet autres vendeurs (performative $reject$).

5 Exemple d'illustration

Nous présentons un exemple qui illustre une enchère où un acheteur négocie avec 5 vendeurs $(V_1, \dots, V_5)$ l'achat d'un produit décrit par 2 critères c_1 et c_2. Le point d'aspiration est $a = (80{,}50)$ et le point d'exigence $e = (20{,}5)$. L'incrément ε est fixé à 0.03. A la première étape, les vendeurs formulent les propositions respectives du tableau ci-dessous.

vendeur	valorisation	deviation
V_1	(96,7)	0.57
V_2	(50,55)	0.41
V_3	(94,7)	0.57
V_4	(85,20)	0.40
V_5	(23,82)	0.78

TAB. 2 – *Première itération*

Les points $Ideal$ et $antiIdeal$ sont déduits de l'ensemble des propositions :

$ideal = (96{,}82)$
$antiIdeal = (23{,}7)$

D’où les valeurs:

$\lambda_1 = 1/(96 - 23) = 1/73$
$\lambda_2 = 1/(82 - 7) = 1/75.$

La proposition $b_4^1 = (85{,}20)$ est la meilleure avec $deviation(a{,}b_4^1) = 0.4$. Les niveaux d’exigence de la deuxième étape sont calculés à l’aide de la relation (5):

$e_1^2 = 80 - (0.4 - \varepsilon)/\lambda_1 = 53$
$e_2^2 = 50 - (0.4 - \varepsilon)/\lambda_2 = 22.$

t	e^t	$dev(a{,}e^t)$	vendeurs	m^t	$dev(a{,}m^t)$
1	$(20{,}5)$	0.82	$V_1{,}V_2{,}V_3{,}V_4{,}V_5$	$(85{,}20)$	0.4
2	$(53{,}22)$	0.37	$V_1{,}V_3{,}V_4{,}V_5$	$(75{,}40)$	0.13
3	$(72{,}42)$	0.10	$V_1{,}V_3{,}V_4$	$(73{,}45)$	0.096
4	$(75{,}45)$	0.066	$V_3{,}V_4$	$(77{,}46)$	0.053
5	$(78{,}48)$	0.023	V_3		

TAB. 3 – *Etapes de l’enchère*

Les enchères se déroulent en 5 étapes et se terminent sur un accord avec le vendeur V_3 pour la proposition $b = (77{,}46)$. Le tableau 3 présente la suite des points d’exigence (colonne e^t) ainsi que les meilleures propositions (colonne m^t).

vendeur v_i	meilleure proposition $m(v_i)$	$deviation(a{,}m(v_i))$
V_1	(74, 50)	0.08
V_2	(52, 52)	0.38
V_3	(83, 60)	-0.04
V_4	(76, 48)	0.055
V_5	(70, 57)	0.13

TAB. 4 – *Meilleures propositions des vendeurs*

5.1 Interprétation

Le déroulement de l`enchère peut être interprété en considérant les meilleures propositions de chaque vendeur (voir Tableau 4) et les points d’exigence successivement définis (voir Tableau 3). A chaque étape $t$, un vendeur peut fournir une proposition quand sa meilleure proposition est meilleure que le point d`exigence. A la première étape, tous les vendeurs peuvent répondre. Le vendeur V_2 abandonne à l’étape 2 quand $deviation(a{,}e^2)$ *= 0.37<0.38.* Le vendeur V_5 abandonne à l’étape 3 quand $deviation(a{,}e^3) = 0.10 < 0.13$, le vendeur V_1 à l’étape 4 et le vendeur V_4 à l’étape 5. Le vendeur V_3 gagne l`enchère avec une proposition qui est légèrement meilleure que la meilleure du vendeur V_4.

6 Conclusions et perspectives

Nous avons présenté dans cet article un mécanisme de négociation multicritère pour le commerce électronique basé sur des points d'aspiration et d'exigence. Le point d'aspiration exprime les valeurs souhaitées sur chaque attribut décrivant le produit à acheter. Un point d'exigence représente les valeurs minimales acceptables sur chaque critère d'une proposition. La méthode d'agrégation multicritère utilise la définition d'une déviation au point d'aspiration pour classer les propositions. Une enchère est conduite par des points d'exigence. La définition des points d'exigence force les agents vendeurs à améliorer leur proposition sur tous les critères sans compensation possible. Elle assure la progression de l'enchère vers le point d'aspiration de l'acheteur, ainsi que l'amélioration graduelle des propositions jusqu'à la fin de l'enchère. Elle permet, en outre, à l'agent acheteur de maintenir privé son point d'aspiration et son modèle d'agrégation. Cependant, la définition des niveaux d'aspiration est déterminante dans la conduite des enchères et une limitation apparaît quand le point d'aspiration défini par l'acheteur se trouve loin des propositions réelles du marché. La proposition gagnante reste alors éloignée du point d'aspiration. Dans ce cas, il vaudrait mieux laisser à l'acheteur la possibilité de reformuler son point d'aspiration. Une autre alternative serait de prévoir une phase de prospection précédant les enchères afin d'aider l'acheteur dans la formulation de son point d'aspiration en fonction de l'offre du marché.

Références

[1] S. Aknine, M.J. Bellosta, K. Hamdouni, S. Kornman, and S. Pinson. A many-to-many negotiation protocol for electronic commerce with risk commitment strategies. In *International Conference on Electronic Commerce Research (ICECR-5)*, 2002.

[2] M.J. Bellosta, I. Brigui, S. Kornman, and D. Vanderpooten. A multi-criteria model for electronic auctions. In *19th ACM Symposium on Applied Computing*, page A paraître, 2004.

[3] M. Bichler. An experimental analysis of multi-attribute auctions. *Decision Support Systems*, 29:249–268, 2001.

[4] M. Bichler, M. Kaukal, and A. Segev. Multi-attribute auctions for electronic procurement. In *First IBM IAC Workshop on Internet Based Negotiation Technologies*, volume 44, pages 291–301, 2002.

[5] A. Chavez and P. Maes. Kasbah: An agent marketplace for buying and selling goods. In *First International Conference on the Practical Application of Inteligent Agents and Multi-Agent Technology*, London, Great Britain, 1996.

[6] S. De Vries and R. Vohra. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 2003.

[7] R.H. Guttman, A.G. Moukas, and P. Maes. Agent-mediated electronic commerce : A survey. *Knowledge Engineering Review*, 1998.

[8] N.R. Jennings, P. Faratin, A.R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: prospects, method and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.

[9] K.Y. Lee, J.S. Yun, and G.S. Jo. Mocaas: auction agent system using a collaborative mobile agent in electronic commerce. *Expert System with Applications*, 24:183–187, 2003.

[10] J. Morris and P. Maes. Sardine: An agent-facilitated airline ticket bidding system. In *4th International Conference on Autonomous Agents (Agents 2000)*, Barcelone, Espagne, 2000.

[11] E. Oliveira, J.M. Fonsesca, and A. Steiger-Garçao. Multi-criteria negotiation in multi-agent systems. In *1st International Workshop of Central and Eastern Europe on Multi-agent Systems (CEEMAS'99)*, St. Petersbourg, June 1999.

[12] D.C. Parkes and Ungar L. Preventing strategic manipulation in iterative auctions: Proxy-agents and price-adjusment. In *Seventeenth National Conference on Artificial Intelligence (AAAI'00)*, Austin, USA, 2000.

[13] T.W. Sandholm. Approaches to winner determination in combinatorial auctions. *Decision Support Systems*, 28:165–176, 2000.

[14] R.E. Steuer. *Multiple criteria optimization: theory, computation, and application*. Wiley, New York, 1986.

[15] P. Vallin and D. Vanderpooten. *Aide à la décision : une approche par les cas*. Ellipses, Paris, 2002.

[16] W. Vickrey. Counterspeculation auctions and competitive sealed tenders. *Journal of finance*, 16:8–37, 1961.

[17] N Vulkan and N.R Jennings. Efficient mechanisms for the supply of services in multi-agent environments. *Decision Support Systems*, 28:5–19, 2000.

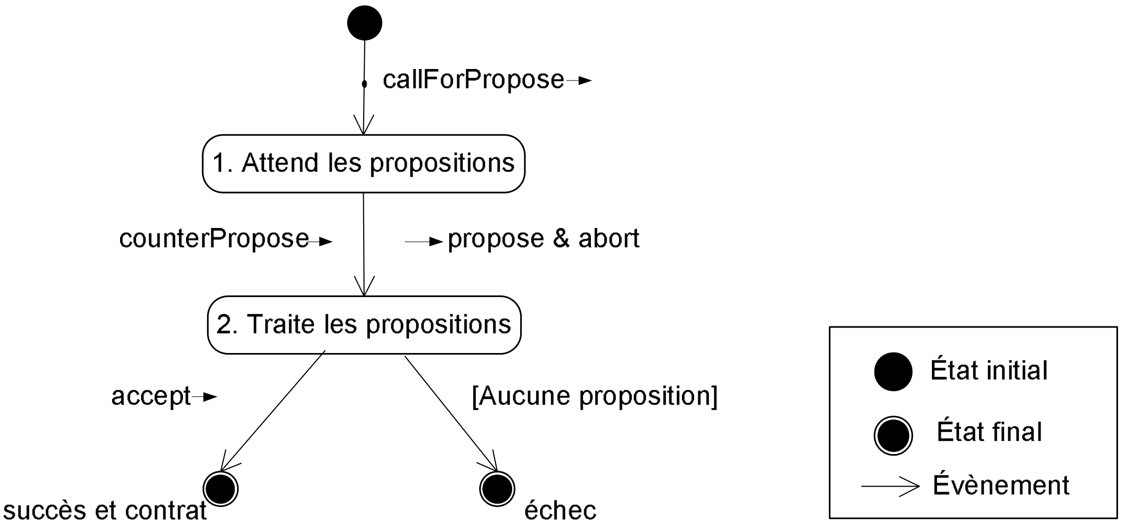


FIG. 1 – *Graphe d'états*

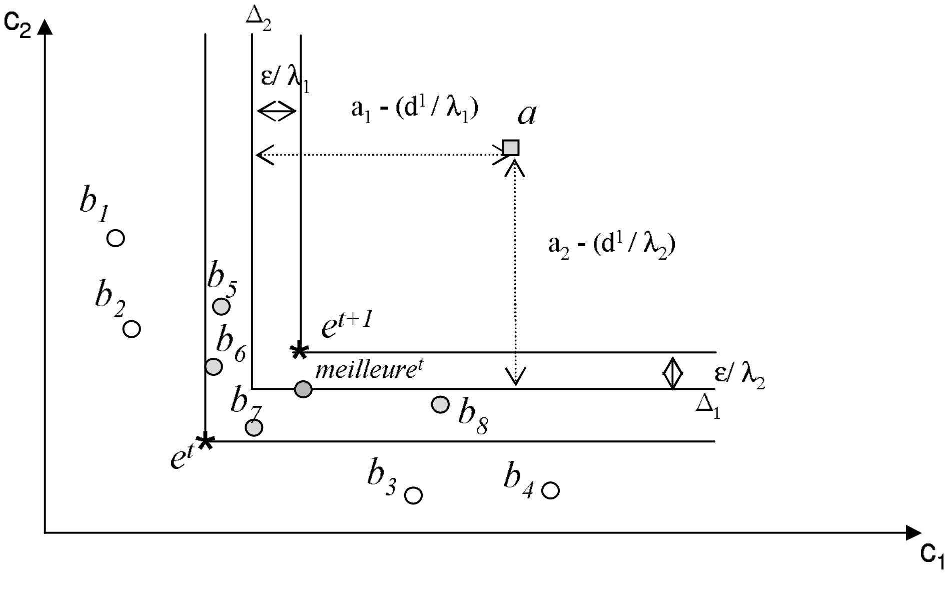


FIG. 2 – *Détermination du point d'exigence*

Conjoint measurement tools for MCDM
A brief introduction

Denis Bouyssou[*], Marc Pirlot[†]

Résumé

Ce texte vise à introduire aux principales techniques de mesurage conjoint utiles en analyse multicritère. L'accent est principalement mis sur le modèle central des fonctions de valeur additives. On présente brièvement ses fondements ainsi que diverses techniques permettant de le mettre en œuvre. On présente ensuite diverses extensions de ce modèle, par exemples des modèles non additifs et/ou tolérant la présence d'intransitivités.

Mots-clefs : Mesurage conjoint, Fonctions de valeur additives, Modélisation des préférences

Abstract

This paper offers a brief and nontechnical introduction to the use of conjoint measurement in multiple criteria decision making. The emphasis is on the, central, additive value function model. We outline its axiomatic foundations and present various possible assessment techniques to implement it. Some extensions of this model, e.g. nonadditive models or models tolerating intransitive preferences are then briefly reviewed.

Key words : Conjoint Measurement, Additive Value Function, Preference Modelling

[*] LAMSADE, Université Paris-Dauphine, 75775 Paris cedex 16, France. bouyssou@lamsade.dauphine.fr

[†] Faculté Polytechnique de Mons, 9, rue de Houdain, B-7000 Mons, Belgique. marc.pirlot@fpms.ac.be

1 Introduction and motivation

Conjoint measurement is a set of tools and results first developed in Economics [44] and Psychology [141] in the beginning of the ‘60s. Its, ambitious, aim is to provide measurement techniques that would be adapted to the needs of the Social Sciences in which, most often, multiple dimensions have to be taken into account.

Soon after its development, people working in decision analysis realized that the techniques of conjoint measurement could also be used as tools to structure preferences [51, 165]. This is the subject of this paper which offers a brief and nontechnical introduction to conjoint measurement models and their use in multiple criteria decision making. More detailed treatments may be found in [63, 79, 121, 135, 209]. Advanced references include [58, 129, 211].

1.1 Conjoint measurement models in decision theory

The starting point of most works in decision theory is a binary relation $\succsim$ on a set A of objects. This binary relation is usually interpreted as an “at least as good as” relation between alternative courses of action gathered in A.

Manipulating a binary relation can be quite cumbersome as soon as the set of objects is large. Therefore, it is not surprising that many works have looked for a *numerical representation* of the binary relation $\succsim$. The most obvious numerical representation amounts to associate a real number $V(a)$ to each object $a \in A$ in such a way that the comparison between these numbers faithfully reflects the original relation $\succsim$. This leads to defining a real-valued function V on A, such that:

$$a \succsim b \Leftrightarrow V(a) \geq V(b), \tag{1}$$

for all $a, b \in A$. When such a numerical representation is possible, one can use V instead of $\succsim$ and, e.g. apply classical optimization techniques to find the most preferred elements in A given $\succsim$. We shall call such a function V a *value function*.

It should be clear that not all binary relations $\succsim$ may be represented by a value function. Condition (1) imposes that $\succsim$ is complete (i.e. $a \succsim b$ or $b \succsim a$, for all $a, b \in A$) and transitive (i.e. $a \succsim b$ and $b \succsim c$ imply $a \succsim c$, for all $a, b, c \in A$). When A is finite or countably infinite, it is well-known [58, 129] that these two conditions are, in fact, not only necessary but also sufficient to build a value function satisfying (1).

Remark 1

The general case is more complex since (1) implies, for instance, that there must be “enough” real numbers to distinguish objects that have to be distinguished. The necessary and sufficient conditions for (1) can be found in [58, 129]. An advanced treatment is [13].

Sufficient conditions that are well-adapted to cases frequently encountered in Economics can be found in [42, 45]; see [34] for a synthesis. •

It is vital to note that, when a value function satisfying (1) exists, it is by no means unique. Taking any increasing function ϕ on $\mathbb{R}$, it is clear that $\phi \circ V$ gives another acceptable value function. A moment of reflection will convince the reader that only such transformations are acceptable and that if V and U are two real-valued functions on A satisfying (1), they must be related by an increasing transformation. In other words, a value function in the sense of (1) defines an *ordinal scale*.

Ordinal scales, although useful, do not allow the use of sophisticated assessment procedures, i.e. of procedures that allow an analyst to assess the relation $\succsim$ through a structured dialogue with the decision-maker. This is because the knowledge that $V(a) \geq V(b)$ is strictly equivalent to the knowledge of $a \succsim b$ and no inference can be drawn from this assertion besides the use of transitivity.

It is therefore not surprising that much attention has been devoted to numerical representations leading to more constrained scales. Many possible avenues have been explored to do so. Among the most well-known, let us mention:

- the possibility to compare *probability distributions* on the set A [58, 207]. If it is required that, not only (1) holds but that the numbers attached to the objects should be such that their expected values reflect the comparison of probability distributions on the set of objects, a much more constrained numerical representation clearly obtains,
- the introduction of "preference difference" comparisons of the type: the difference between a and b is larger than the difference between c and d, see [44, 81, 123, 129, 159, 180, 199]. If it is required that, not only (1) holds, but that the differences between numbers also reflect the comparisons of preference differences, a more constrained numerical representation obtains.

When objects are evaluated according to several dimensions, i.e. when $\succsim$ is defined on a product set, new possibilities emerge to obtain numerical representations that would specialize (1). The purpose of conjoint measurement is to study such kinds of models.

There are many situations in decision theory which call for the study of binary relations defined on product sets. Among them let us mention:

- *Multiple criteria decision making* using a preference relation comparing alternatives evaluated on several attributes [16, 121, 162, 173, 209],
- *Decision under uncertainty* using a preference relation comparing alternatives evaluated on several states of nature [68, 107, 177, 184, 210, 211],

- *Consumer theory* manipulating preference relations for bundles of several goods [43],
- *Intertemporal decision making* using a preference relation between alternatives evaluated at several moments in time [121, 125, 126],
- *Inequality measurement* comparing distributions of wealth across several individuals [5, 17, 18, 217].

The purpose of this paper is to give an introduction to the main models of conjoint measurement useful in multiple criteria decision making. The results and concepts that are presented may however be of interest in all of the afore-mentioned areas of research.

Remark 2

Restricting ourselves to applications in multiple criteria decision making will not allow us to cover every aspect of conjoint measurement. Among the most important topics left aside, let us mention: the introduction of statistical elements in conjoint measurement models [54, 108] and the test of conjoint measurement models in experiments [135]. •

Given a binary relation $\succsim$ on a product set $X = X_1 \times X_2 \times \cdots \times X_n$, the theory of conjoint measurement consists in finding conditions under which it is possible to build a convenient numerical representation of $\succsim$ and to study the uniqueness of this representation. The central model is the *additive value function* model in which:

$$x \succsim y \Leftrightarrow \sum_{i=1}^{n} v_i(x_i) \geq \sum_{i=1}^{n} v_i(y_i) \tag{2}$$

where v_i are real-valued functions, called *partial value functions*, on the sets X_i and it is understood that $x = (x_1, x_2, \ldots , x_n)$ and $y = (y_1, y_2, \ldots, y_n)$. Clearly if $\succsim$ has a representation in model (2), taking any common increasing transformation of the v_i will *not* lead to another representation in model (2).

Specializations of this model in the above-mentioned areas give several central models in decision theory:

- The Subjective Expected Utility model, in the case of decision-making under uncertainty,
- The discounted utility model for dynamic decision making,
- Inequality measures *à la* Atkinson/Sen in the area of social welfare.

The axiomatic analysis of this model is now quite firmly established [44, 129, 211]; this model forms the basis of many decision analysis techniques [79, 121, 209, 211]. This is studied in sections 3 and 4 after we introduce our main notation and definitions in section 2.

Remark 3

One possible objection to the study of model (2) is that the choice of an *additive* model seems arbitrary and restrictive. It should be observed here that the functions v_i will precisely be assessed so that additivity holds. Furthermore, the use of a simple model may be seen as an advantage in view of the limitations of the cognitive abilities of most human beings.

It is also useful to notice that this model can be reformulated so as to make addition disappear. Indeed if there are partial value functions v_i such that (2) holds, it is clear that $V = \sum_{i=1}^{n} v_i$ is a value function satisfying (1). Since V defines an ordinal scale, taking the exponential of V leads to another valid value function W. Clearly W has now a multiplicative form:

$$x \succsim y \Leftrightarrow W(x) = \prod_{i=1}^{n} w_i(x_i) \geq W(y) = \prod_{i=1}^{n} w_i(y_i).$$

where $w_i(x_i) = \mathrm{e}^{v_i(x_i)}$.

The reader is referred to [50, 209] for the study of situations in which V defines a scale that is more constrained than an ordinal scale, e.g. because it is supposed to reflect preference differences or because it allows to compute expected utilities. In such cases, the additive form (2) is no more equivalent to the multiplicative form considered above. •

In section 5 we present a number of extensions of this model going from nonadditive representations of transitive relations to model tolerating intransitive indifference and, finally, nonadditive representations of nontransitive relations.

Remark 4

In this paper, we shall restrict our attention to the case in which alternatives may be evaluated on the various attributes without risk or uncertainty. Excellent overviews of these cases may be found in [121, 209]; recent references include [142, 150]. •

Before starting our study of conjoint measurement oriented towards MCDM, it is worth recalling that conjoint measurement aims at establishing measurement models in the Social Sciences. To many, the very notion of “measurement in the Social Sciences” may appear contradictory. It may therefore be useful to briefly consider how the notion of measurement can be modelled in Physics, an area in which the notion of “measurement” seems to arise quite naturally, and to explain how a “measurement model” may indeed be useful in order to structure preferences.

1.2 An aside: measuring length

Physicists usually take measurement for granted and are not particularly concerned with the technical and philosophical issues it raises (at least when they work within the realm of Newtonian Physics). However, for a Social Scientist, these question are of utmost importance. It may thus help to have an idea of how things appear to work in Physics before tackling more delicate cases.

Suppose that you are on a desert island and that you want to "measure" the length of a collection of rigid straight rods. Note that we do not discuss here the "pre-theoretical" intuition that "length" is a property of these rods that can be measured, as opposed, say, to their softness or their beauty.

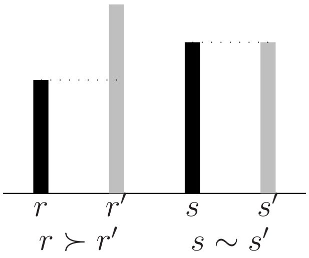


Figure 1: Comparing the length of two rods.

A first simple step in the construction of a measure of length is to place the two rods side by side in such a way that one of their extremities is at the same level (see Figure 1). Two things may happen: either the upper extremities of the two rods coincide or not. This seems to be the simplest way to devise an experimental procedure leading to the discovery of which rod "has more length" than the other. Technically, this leads to defining two binary relations $\succ$ and $\sim$ on the set of rods in the following way:

$$r \succ r' \text{ when the extremity of } r \text{ is higher than the extremity of } r',$$
$$r \sim r' \text{ when the extremities of } r \text{ and } r' \text{ are at the same level,}$$

Clearly, if length is a quality of the rods that can be measured, it is expected that these pairwise comparisons are somehow consistent, e.g.,

- if $r \succ r'$ and $r' \succ r''$, it should follow that $r \succ r''$,
- if $r \sim r'$ and $r' \sim r''$, it should follow that $r \sim r''$,
- if $r \sim r'$ and $r' \succ r''$, it should follow that $r \succ r''$.

Although quite obvious, these consistency requirements are stringent. For instance, the second and the third conditions are likely to be violated if the experimental procedure involves some imprecision, e.g if two rods that slightly differ in length are nevertheless judged “equally long”. They represent a form of *idealization* of what could be a perfect experimental procedure.

With the binary relations $\succ$ and $\sim$ at hand, we are still rather far from a full-blown measure of length. It is nevertheless possible to assign numbers to each of the rods in such a way that the comparison of these numbers reflects what has been obtained experimentally. When the consistency requirements mentioned above are satisfied, it is indeed generally possible to build a real-valued function Φ on the set of rods that would satisfy:

$$r \succ r' \Leftrightarrow \Phi(r) > \Phi(r') \text{ and}$$
$$r \sim r' \Leftrightarrow \Phi(r) = \Phi(r').$$

If the experiment is costly or difficult to perform, such a numerical assignment may indeed be useful because it summarizes, once for all, what has been obtained in experiments. Clearly there are many possible ways to assign numbers to rods in this way. Up to this point, they are equally good for our purposes. The reader will easily check that defining $\succsim$ as $\succ$ or $\sim$, the function Φ is noting else than a “value function” for length: any increasing transformation may therefore be applied to Φ.

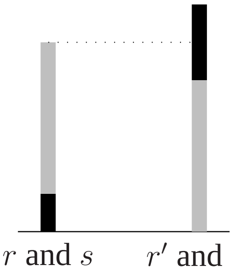


Figure 2: Comparing the length of composite rods.

The next major step towards the construction of a measure of length is the realization that it is possible to form new rods by simply placing two or more rods “in a row”, i.e. you may *concatenate* rods. From the point of view of length, it seems obvious to expect this concatenation operation $\circ$ to be “commutative” ($r \circ s$ has the same length as $s \circ r$) and associative ($(r \circ s) \circ t$ has the same length as $r \circ (s \circ t)$).

You clearly want to be able to measure the length of these composite objects and you can always include them in our experimental procedure outlined above (see Figure 2). Ideally, you would like your numerical assignment Φ to be somehow compatible with the concatenation operation: knowing the numbers assigned to two rods, you want to be able to deduce the number assigned to their concatenation. The most obvious way to achieve that is to require that the numerical assignment of a composite object can be deduced by addition from the numerical assignments of the objects composing it, i.e. that

$$\Phi(r \circ r') = \Phi(r) + \Phi(r').$$

This clearly places many additional constraints on the results of your experiment. An obvious one is that $\succ$ and $\sim$ should be compatible with the concatenation operation $\circ$, e.g.

$$r \succ r' \text{ and } t \sim t' \text{ should lead to } r \circ t \succ r' \circ t'.$$

These new constraints may or may not be satisfied. When they are, the usefulness of the numerical assignment Φ is even more apparent: a simple arithmetic operation will allow to infer the result of an experiment involving composite objects.

Let us take a simple example. Suppose that you have 5 rods $r_1, r_2, \ldots, r_5$ and that, because space is limited, you can only concatenate at most two rods and that not all concatenations are possible. Let us suppose, for the moment, that you do not have much technology available so that you may only experiment using *different* rods. You may well collect the following information, using obvious notation exploiting the transitivity of $\succ$ which holds in this experiment,

$$r_1 \circ r_5 \succ r_3 \circ r_4 \succ r_1 \circ r_2 \succ r_5 \succ r_4 \succ r_3 \succ r_2 \succ r_1.$$

Your problem is then to find a numerical assignment Φ to rods such that using an addition operation, you can infer the numerical assignment of composite objects consistently with your observations. Let us consider the following three assignments:

	Φ	Φ'	Φ''
r_1	14	10	14
r_2	15	91	16
r_3	20	92	17
r_4	21	93	18
r_5	28	100	29

These three assignments are equally valid to reflect the comparisons of single rods. Only the first and the third allow to capture the comparisons of composite objects that were performed. Note that, going from Φ to Φ'' does not involve just changing the "unit of measurement": since $\Phi(r_1) = \Phi''(r_1)$ this would imply that $\Phi = \Phi''$, which is clearly false.

Such numerical assignments have limited usefulness. Indeed, it is tempting to use them to predict the result of comparisons that we have not been able to perform. But this turns out to be quite disappointing: using Φ you would conclude that $r_2 \circ r_3 \sim r_1 \circ r_4$ since $\Phi(r_2)+\Phi(r_3) = 15+20 = 35 = \Phi(r_1)+\Phi(r_4)$, but, using Φ'', you would conclude that $r_2 \circ r_3 \succ r_1 \circ r_4$ since $\Phi''(r_2) + \Phi''(r_3) = 16 + 17 = 33$ while $\Phi''(r_1) + \Phi''(r_4) = 14 + 18 = 32$.

Intuitively, “measuring” calls for some kind of a *standard* (e.g. the “Mètre-étalon” that can be found in the Bureau International des Poids et Mesures in Sèvres, near Paris). This implies choosing an appropriate “standard” rod *and* being able to prepare perfect copies of this standard rod (we say here “appropriate” because the choice of a standard should be made in accordance with the lengths of the objects to be measured: a tiny or a huge standard will not facilitate experiments). Let us call s_0 the standard rod. Let us suppose that you have been able to prepare a large number of perfect copies $s_1, s_2, \ldots$ of s_0. We therefore have:

$$s_0 \sim s_1, s_0 \sim s_2, s_0 \sim s_3, \ldots$$

Let us also agree that the length of s_0 is 1. This is your, arbitrary, unit of length. How can you use s_0 and its perfect copies so as to determine unambiguously the length of any other (simple or composite) object? Quite simply, you may prepare a “standard sequence of length n”, $S(n) = s_1 \circ s_2 \circ \ldots \circ s_{n-1} \circ s_n$, i.e. a composite object that is made by concatenating n perfect copies of our standard rod s_0. The length of a standard sequence of length n is exactly n since we have concatenated n objects that are perfect copies of the standard rod of length 1. Take any rod r and let us compare r with several standard sequences of increasing length: $S(1), S(2), \ldots$

Two cases may arise. There may be a standard sequence $S(k)$ such that $r \sim S(k)$. In that case, we know that the number $\Phi(r)$ assigned to r must be exactly k. This is unlikely however. The most common situation is that we will find two consecutive standard sequences $S(k-1)$ and $S(k)$ such that $r \succ S(k-1)$ and $S(k) \succ r$ (see Figure 3). This means that $\Phi(r)$ must be such that $k - 1 < \Phi(r) < k$. We seem to be in trouble here since, as before, $\Phi(r)$ is not exactly determined. How can you proceed? This depends on your technology for preparing perfect copies.

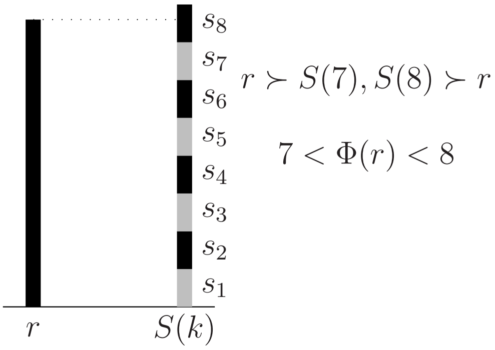


Figure 3: Using standard sequences.

Imagine that you are able to prepare perfect copies not only of the standard rod but also of any object. You may then prepare several copies (r_1, $r_2, \ldots$) of the rod r. You can now compare a composite object made out of two perfect copies of r with your standard sequences $S(1), S(2), \ldots$ As before, you shall eventually arrive at locating $\Phi(r_1 \circ r_2) = 2\Phi(r)$ within an interval of width 1. This means that the interval of imprecision surrounding $\Phi(r)$ has been divided by two. Continuing this process, considering longer and longer sequences of perfect copies of r, you will keep on reducing the width of the interval containing $\Phi(r)$. This means that you can approximate $\Phi(r)$ with any given level of precision. Mathematically, a unique value for $\Phi(r)$ will be obtained using a simple argument.

Supposing that you are in position to prepare perfect copies of any object is a strong technological requirement. When this is not possible, there still exists a way out. Instead of preparing a perfect copy of r you may also try to increase the granularity of your standard sequence. This means building an object t that you would be able to replicate perfectly and such that concatenating t with one of its perfect replicas gives an object that has exactly the length of the standard object s_0, i.e. $\Phi(t) = 1/2$. Considering standard sequences based on t, you will be able to increase by a factor 2 the precision with which we measure the length of r. Repeating the process, i.e. subdividing t, will lead, as before, to a unique limiting value for $\Phi(r)$.

The mathematical machinery underlying the measurement process informally described above (called "extensive measurement") rests on the theory of ordered groups. It is beautifully described and illustrated in [129]. Although the underlying principles are simple, we may expect complications to occur e.g. when not all concatenations are feasible, when there is some level (say the velocity of light if we were to measure speed) that cannot be exceeded or when it comes to relate different measures. See [129, 140, 168] for a detailed treatment.

Clearly, this was an overly detailed and unnecessary complicated description of how length could be measured. Since our aim is to eventually deal with "measurement" in the Social Sciences, it may however be useful to keep the above process in mind. Its basic ingredients are the following:

- well-behaved relations $\succ$ and $\sim$ allowing to compare objects,
- a concatenation operation $\circ$ allowing to consider composite objects,
- consistency requirements linking $\succ$, $\sim$ and $\circ$,
- the ability to prepare perfect copies of some objects in order to build standard sequences.

Basically, conjoint measurement is a quite ingenious way to perform related measurement operations when no concatenation operation is available. This will however require that objects can be evaluated along several dimensions. Before explaining how this might work, it is worth explaining the context in which such measurement might prove useful.

Remark 5

It is often asserted that "measurement is impossible in the Social Sciences" precisely because the Social Scientist has no way to define a concatenation operation. Indeed, it would seem hazardous to try to concatenate the intelligence of two subjects or the pain of two patients (see [56, 106]). Under certain conditions, the power of conjoint measurement will precisely be to provide a means to bypass this absence of readily available concatenation operation when the objects are evaluated on several dimensions.

Let us remark that, even when there seems to be a concatenation operation readily available, it does not always fit the purposes of extensive measurement. Consider for instance an individual expressing preferences for the quantity of the two goods he consumes. The objects therefore take the well structured form of points in the positive orthant of $\mathbb{R}^2$. There seems to be an obvious concatenation operation here: $(x, y) \circ (z, w)$ might simply be taken to be $(x + y, z + w)$. However a fairly rational person, consuming pants and jackets, may indeed prefer $(3, 0)$ (3 pants and no jacket) to $(0, 3)$ (no pants and 3 jackets) but at the same time prefer $(3, 3)$ to $(6, 0)$. This implies that these preferences cannot be explained by a measure that would be additive with respect to the concatenation operation consisting in adding the quantities of the two goods consumed. Indeed $(3, 0) \succ (0, 3)$ implies $\Phi(3, 0) > \Phi(0, 3)$, which implies $\Phi(3, 0) + \Phi(3, 0) > \Phi(0, 3) + \Phi(3, 0)$. Additivity with respect to concatenation should then imply that $(3, 0) \circ (3, 0) \succ (0, 3) \circ (3, 0)$, that is $(6, 0) \succ (3, 3)$.

1.3 An example: Even swaps

The even swaps technique described and advocated in [120, 121, 165] is a simple way to deal with decision problems involving several attributes that does not have recourse to a formal representation of preferences, which will be the subject of conjoint measurement. Because this technique is simple and may be quite useful, we describe it below using the same example as in [120]. This will also allow to illustrate the type of problems that are dealt with in decision analysis applications of conjoint measurement.

Example 6 (Even swaps technique)

A consultant considers renting a new office. Five different locations have been identified after a careful consideration of many possibilities, rejecting all those that do not meet a number of requirements.

His feeling is that five distinct characteristics, we shall say five attributes, of the possible locations should enter into his decision: his daily commute time (expressed in minutes), the ease of access for his clients (expressed as the percentage of his present clients living close to the office), the level of services offered by the new office (expressed on an ad hoc scale with three levels: A (all facilities available), B (telephone and fax), C (no facilities)), the size of the office expressed in square feet, and the monthly cost expressed in dollars.

The evaluation of the five offices is given in Table 1. The consultant has well-defined preferences on each of these attributes, independently of what is happening on the other attributes. His preference increases with the level of access for his clients, the level of services of the office and its size. It decreases with commute time and cost. This gives a first easy way to compare alternatives through the use of *dominance*.

	a	b	c	d	e
Commute	45	25	20	25	30
Clients	50	80	70	85	75
Services	A	B	C	A	C
Size	800	700	500	950	700
Cost	1850	1700	1500	1900	1750

Table 1: Evaluation of the 5 offices on the 5 attributes.

An alternative y is dominated by an alternative x if x is at least as good as y on *all* attributes while being strictly better for at least one attribute. Clearly dominated alternatives are not candidate for the final choice and may, thus, be dropped from consideration. The reader will easily check that, on this example, alternative b dominates alternative e: e and b have similar size but b is less expensive, involves a shorter commute time, an easier access to clients and a better level of services. We may therefore forget about alternative e. This is the only case of “pure dominance” in our table. It is however easy to see that d is “close” to dominating a, the only difference in favor of a being on the cost attribute (50 $ per month). This is felt more than compensated by the differences in favor of d on all other attributes: commute time (20 minutes), client access (35 %) and size (150 sq. feet).

Dropping all alternatives that are not candidate for choice, this initial investigation allows to reduce the problem to:

	b	c	d
Commute	25	20	25
Clients	80	70	85
Services	B	C	A
Size	700	500	950
Cost	1700	1500	1900

A natural way to proceed is then to assess tradeoffs. Observe that all alternatives but c have a common evaluation on commute time. We may therefore ask the consultant, starting with office c, what gain on client access would compensate a loss of 5 minutes on commute time. We are looking for an alternative c' that would be evaluated as follows:

	c	c'
Commute	20	$\mathbf{25}$
Clients	70	$\mathbf{70}+\delta$
Services	C	C
Size	500	500
Cost	1500	1500

and judged indifferent to c. Although this is not an easy question, it is clearly crucial in order to structure preferences.

Remark 7

In this paper, we do not consider the possibility of lexicographic preferences, in which such tradeoffs do not occur, see [59, 60, 160]. Lexicographic preferences may also be combined with the possibility of "local" tradeoffs, see [22, 64, 136]. •

Remark 8

Since tradeoffs questions may be difficult, it is wise to start with an attribute requiring few assessments (in the example, all alternatives but one have a common evaluation on commute time). Clearly this attribute should be traded against one with an underlying "continuous" structure (cost, in the example). •

Suppose that the answer is that for $\delta = 8$, it is reasonable to assume that c and c' would be indifferent. This means that the decision table can be reformulated as follows:

	b	c'	d
Commute	25	25	25
Clients	80	78	85
Services	B	C	A
Size	700	500	950
Cost	1700	1500	1900

It is then apparent that all alternatives have a similar evaluation on the first attribute which, therefore, is not useful to discriminate between alternatives and may be forgotten. The reduced decision table is as follows:

	b	c'	d
Clients	80	78	85
Services	B	C	A
Size	700	500	950
Cost	1700	1500	1900

There is no case of dominance in this reduced table. Therefore further simplification calls for the assessment of new tradeoffs. Using cost as the reference attribute, we then proceed to "neutralize" the service attribute. Starting with office c', this means asking for the increase in monthly cost that the consultant would just be prepared to pay to go from level "C" of service to level "B". Suppose that this increase is roughly 250 \$. This defines alternative c''. Similarly, starting with office d we ask for the reduction of cost that would exactly compensate a reduction of services from "A" to "B". Suppose that the answer is 100 \$ a month, which defines alternative d'. The decision table is reshaped as:

	b	c''	d'
Clients	80	78	85
Services	B	$\mathbf{B}$	$\mathbf{B}$
Size	700	500	950
Cost	1700	**1750**	**1800**

We may forget about the second attribute which does not discriminate any more between alternatives. When this is done, it is apparent that c'' is dominated by b and can be suppressed. Therefore, the decision table at this stage looks like the following:

	b	d'
Clients	80	85
Size	700	950
Cost	1700	1800

Unfortunately, this table reveals no case of dominance. New tradeoffs have to be assessed. We may now ask, starting with office b, what additional cost the consultant would be ready to incur to increase its size by 250 square feet. Suppose that the rough answer is 250 \$ a month, which defines b'. We are now facing the following table:

	b'	d'
Clients	80	85
Size	$\mathbf{950}$	950
Cost	$\mathbf{1950}$	1800

Attribute size may now be dropped from consideration. But, when this is done, it is clear that d' dominates b'. Hence it seems obvious to recommend office d as the final choice. ◇

The above process is simple and looks quite obvious. If this works, why be interested at all in "measurement" if the idea is to help someone to come up with a decision?

First observe that in the above example, the set of alternatives was relatively small. In many practical situations, the set of objects to compare is much larger than the set of alternatives in our example. Using the even swaps technique could then require a considerable number of difficult tradeoff questions. Furthermore, as the output of the technique is not a preference model but just the recommendation of an alternative in a given set, the appearance of new alternatives (e.g. because a new office is for rent) would require starting a new round of questions. This is likely to be highly frustrating. Finally, the informal even swaps technique may not be well adapted to the, many, situations, in which the decision under study takes place in a complex organizational environment. In such situations, having a formal model to be able to communicate and to convince is an invaluable asset. Such a model will furthermore allow to conduct extensive sensitivity analysis and, hence, to deal with imprecision both in the evaluations of the objects to compare and in the answers to difficult questions concerning tradeoffs.

This clearly leaves room for a more formal approach to structure preferences. But where can "measurement" be involved in the process? It should be observed that, beyond surface, there are many analogies between the even swaps process and the measurement of length considered above.

First, note that, in both cases, objects are compared using binary relations. In the measurement of length, the binary relation $\succ$ reads "is longer than". Here it reads "is preferred to". Similarly, the relation $\sim$ reading before "has equal length" now reads "is indifferent to". We supposed in the measurement of length process that $\succ$ and $\sim$ would nicely combine in experiments: if $r \succ r'$ and $r' \sim r''$ then we should observe that $r \succ r''$. Implicitly, a similar hypothesis was made in the even swaps technique. To realize that this is the case, it is worth summarizing the main steps of the argument.

We started with Table 1. Our overall recommendation was to rent office d. This means that we have reasons to believe that d is preferred to all other potential locations, i.e. $d \succ a$, $d \succ b$, $d \succ c$, and $d \succ e$. How did we arrive logically at such a conclusion?

Based on the initial table, using dominance and quasi-dominance, we concluded that b was preferable to e and that d was preferable to a. Using symbols, we have $b \succ e$ and $d \succ a$. After assessing some tradeoffs, we concluded, using dominance, that $b \succ c''$. But remember, c'' was built so as to be indifferent to c' and, in turn, c' was built so as to be indifferent to c. That is, we have $c'' \sim c'$ and $c' \sim c$. Later, we built an alternative d' that is indifferent to d ($d \sim d'$) and an alternative b' that is indifferent to b ($b \sim b'$). We then concluded, using dominance, that d' was preferable to b' ($d' \succ b'$). Hence, we know that:

$$\begin{gathered} d \succ a, b \succ e, \\ c'' \sim c', c' \sim c, b \succ c'', \\ d \sim d', b \sim b', d' \succ b'. \end{gathered}$$

Using the consistency rules linking $\succ$ and $\sim$ that we considered for the measurement of length, it is easy to see that the last line implies $d \succ b$. Since $b \succ e$, this implies $d \succ e$. It remains to show that $d \succ c$. But the second line leads to, combining $\succ$ and $\sim$, $b \succ c$. Therefore $d \succ b$ leads to $d \succ c$ and we are home. Hence, we have used the same properties for preference and indifference as the properties of "is longer than" and "has equal length" that we hypothesized in the measurement of length.

Second it should be observed that expressing tradeoffs leads, indirectly, to equating the "length" of "preference intervals" on different attributes. Indeed, remember how c' was constructed above: saying that c and c' are indifferent more or less amounts to saying that the interval $[25, 20]$ on commute time has exactly the same "length" as the interval $[70, 78]$ on client access. Consider an alternative f that would be identical to c except that it has a client access at 78%. We may again ask which increase in client access would compensate a loss of 5 minutes on commute time. In a tabular form we are now comparing the following two alternatives:

	f	f'
Commute	20	25
Clients	78	$78 + \delta$
Services	C	C
Size	500	500
Cost	1500	1500

Suppose that the answer is that for $\delta = 10$, f and f' would be indifferent. This means that the interval $[25, 20]$ on commute time has exactly the same length as the interval $[78, 88]$ on client access. Now, we know that the preference intervals $[70, 78]$ and $[78, 88]$ have the same "length". Hence, tradeoffs provide a means to equate two preference intervals on the same attribute. This brings us quite close to the construction of standard sequences. This, we shall shortly do.

How does this information about the "length" of preference intervals relate to judgements of preference or indifference? Exactly as in the even swaps technique. You can use this measure of "length" modifying alternatives in such a way that they only differ on a single attribute and then use a simple dominance argument.

Conjoint measurement techniques may roughly be seen as a formalization of the even swaps technique that leads to building a numerical model of preferences much in the same way that we built a numerical model for length. This will require assessment procedures that will rest on the same principles as the standard sequence technique used for length. This process of "measuring preferences" is not an easy one. It will however lead to a numerical model of preference that will not only allow us to make a choice within a limited number of alternatives but that can serve as an input of computerized optimization algorithms that will be able to deal with much more complex cases.

2 Definitions and notation

Before entering into the details of how conjoint measurement may work, a few definitions and notation will be needed.

2.1 Binary relations

A *binary relation* $\succsim$ on a set A is a subset of $A \times A$. We write $a \succsim b$ instead of $(a, b) \in \succsim$. A binary relation $\succsim$ on A is said to be:

- *reflexive* if $[a \succsim a]$,
- *complete* if $[a \succsim b$ or $b \succsim a]$,
- *symmetric* if $[a \succsim b] \Rightarrow [b \succsim a]$,
- *asymmetric* if $[a \succsim b] \Rightarrow [Not[b \succsim a]]$,
- *transitive* if $[a \succsim b$ and $b \succsim c] \Rightarrow [a \succsim c]$,
- *negatively transitive* if $[\,Not[\,a \succsim b\,]$ and $Not[\,b \succsim c\,]\,] \Rightarrow\ Not[\,a \succsim c\,]$,

for all $a, b, c \in A$.

The *asymmetric* (resp. *symmetric*) part of $\succsim$ is the binary relation $\succ$ (resp. $\sim$) on A defined letting, for all $a, b \in A$, $a \succ b \Leftrightarrow [a \succsim b$ and $Not(b \succsim a)]$ (resp. $a \sim b \Leftrightarrow [a \succsim b$ and $b \succsim a]$). A similar convention will hold when $\succsim$ is subscripted and/or superscripted.

A *weak order* (resp. an *equivalence relation*) is a complete and transitive (resp. reflexive, symmetric and transitive) binary relation. For a detailed analysis of the use of binary relation as tools for preference modelling we refer to [4, 58, 66, 161, 167, 169]. The weak order model underlies the examples that were presented in the introduction. Indeed, the reader will easily prove the following.

Proposition 9
Let $\succsim$ be a weak order on A. Then:

- $\succ$ *is transitive,*
- $\succ$ *is negatively transitive,*
- $\sim$ *is transitive,*
- $[a \succ b$ *and* $b \sim c] \Rightarrow a \succ c$*,*
- $[a \sim b$ *and* $b \succ c] \Rightarrow a \succ c$*,*

for all $a, b, c \in A$.

2.2 Binary relations on product sets

In the sequel, we consider a set $X = \prod_{i=1}^{n} X_i$ with $n \geq 2$. Elements $x, y, z, \dots$ of X will be interpreted as alternatives evaluated on a set $N = \{1, 2, \dots, n\}$ of attributes. A typical binary relation on X is still denoted as $\succsim$, interpreted as an "at least as good as" preference relation between multi-attributed alternatives with $\sim$ interpreted as indifference and $\succ$ as strict preference.

For any nonempty subset J of the set of attributes N, we denote by X_J (resp. X_{-J}) the set $\prod_{i \in J} X_i$ (resp. $\prod_{i \notin J} X_i$). With customary abuse of notation, (x_J, y_{-J}) will denote the element $w \in X$ such that $w_i = x_i$ if $i \in J$ and $w_i = y_i$ otherwise. When $J = \{i\}$ we shall simply write X_{-i} and (x_i, y_{-i}).

Remark 10
Throughout this paper, we shall work with a binary relation defined on a product set. This setup conceals the important work that has to be done in practice to make it useful:

- the structuring of objectives [3, 15, 16, 117, 118, 119, 157, 163],
- the definition of adequate attributes to measure the attainment of objectives [80, 96, 116, 122, 173, 208, 216],
- the definition of an adequate family of attributes [24, 121, 173, 174, 209],
- the modelling of uncertainty, imprecision and inaccurate determination [23, 27, 121, 171].

The importance of this "preliminary" work should not be forgotten in what follows. •

2.3 Independence and marginal preferences

In conjoint measurement, one starts with a preference relation $\succsim$ on X. It is then of vital importance to investigate how this information makes it possible to define preference relations on attributes or subsets of attributes.

Let $J \subseteq N$ be a nonempty set of attributes. We define the *marginal relation* $\succsim_J$ induced on X_J by $\succsim$ letting, for all $x_J, y_J \in X_J$:

$$x_J \succsim_J y_J \Leftrightarrow (x_J, z_{-J}) \succsim (y_J, z_{-J}), \text{ for all } z_{-J} \in X_{-J},$$

with asymmetric (resp. symmetric) part $\succ_J$ (resp. $\sim_J$). When $J = \{i\}$, we often abuse notation and write $\succsim_i$ instead of $\succsim_{\{i\}}$. Note that if $\succsim$ is reflexive (resp. transitive), the same will be true for $\succsim_J$. This is clearly not true for completeness however.

Definition 11 (Independence)

Consider a binary relation $\succsim$ *on a set* $X = \prod_{i=1}^{n} X_i$ *and let* $J \subseteq N$ *be a nonempty subset of attributes. We say that* $\succsim$ *is independent for* J *if, for all* $x_J, y_J \in X_J$,

$$[(x_J, z_{-J}) \succsim (y_J, z_{-J}), \text{ for some } z_{-J} \in X_{-J}] \Rightarrow x_J \succsim_J y_J.$$

If $\succsim$ *is independent for all nonempty subsets of* N*, we say that* $\succsim$ *is* independent. *If* $\succsim$ *is independent for all subsets containing a single attribute, we say that* $\succsim$ *is* weakly independent.

In view of (2), it is clear that the additive value model will require that $\succsim$ is independent. This crucial condition says that common evaluations on some attributes do not influence preference. Whereas independence implies weak independence, it is well-know that the converse is not true [211].

Remark 12

Under certain conditions, e.g. when X is adequately "rich", it is not necessary to test that a weak order $\succsim$ is independent for J, for *all* $J \subseteq N$ in order to know that $\succsim$ is independent, see [21, 89, 121]. This is often useful in practice. •

Remark 13

Weak independence is referred to as "weak separability" in [211]; in section 5, we use "weak separability" (and "separability") with a different meaning. •

Remark 14

Independence, or at least weak independence, is an almost universally accepted hypothesis in multiple criteria decision making. It cannot be overemphasized that it is easy to find examples in which it is inadequate.

If a meal is described by the two attributes, main course and wine, it is highly likely that most gourmets will violate independence, preferring red wine with beef and white wine with fish. Similarly, in a dynamic decision problem, a preference for variety will often lead to violating independence: you may prefer Pizza to Steak, but your preference for meals today (first attribute) and tomorrow (second attribute) may well be such that (Pizza, Steak) preferred to (Pizza, Pizza), while (Steak, Pizza) is preferred to (Steak, Steak).

Many authors [119, 173, 209] have argued that such failures of independence were almost always due to a poor structuring of attributes (e.g. in our choice of meal example above, preference for variety should be explicitly modelled). •

When $\succsim$ is a weakly independent weak order, marginal preferences are well-behaved and combine so as to give meaning to the idea of dominance that we already encountered. The proof of the following is left to the reader as an easy exercise.

Proposition 15
Let $\succsim$ *be a weakly independent weak order on* $X = \prod_{i=1}^{n} X_i$*. Then:*

- $\succsim_i$ *is a weak order on* X_i*,*
- $[x_i \succsim_i y_i$*, for all* $i \in N] \Rightarrow x \succsim y$*,*
- $[x_i \succsim_i y_i$*, for all* $i \in N$ *and* $x_j \succ_j y_j$ *for some* $j \in N] \;\Rightarrow x \succ y$*,*

for all $x, y \in X$.

3 The additive value model in the "rich" case

The purpose of this section and the following is to present the conditions under which a preference relation on a product set may be represented by the additive value function model (2) and how such a model can be assessed. We begin here with the case that most closely resembles the measurement of length described in section 1.2.

3.1 Outline of theory

When the structure of X is supposed to be "adequately rich", conjoint measurement is a quite clever adaptation of the process that we described in section 1.2 for the measurement of length. What will be measured here are the "length" of preference intervals on an attribute using a preference interval on another attribute as a standard.

3.1.1 The case of two attributes

Consider first the two attribute case. Hence the relation $\succsim$ is defined on a set $X = X_1 \times X_2$. Clearly, in view of (2), we need to suppose that $\succsim$ is an *independent weak order*. Consider two levels $x_1^0, x_1^1 \in X_1$ on the first attribute such that $x_1^1 \succ_1 x_1^0$, i.e. x_1^1 is preferable to x_1^0. This makes sense because, we supposed that $\succsim$ is *independent*. Note also that we shall have to exclude the case in which all levels on the first attribute would be indifferent in order to be able to find such levels.

Choose any $x_2^0 \in X_2$. The, arbitrarily chosen, element $(x_1^0, x_2^0) \in X$ will be our "reference point". The basic idea is to use this reference point and the "unit" on the first attribute given by the reference preference interval $[x_1^0, x_1^1]$ to build a standard sequence on the preference intervals on the second attribute. Hence, we are looking for an element $x_2^1 \in X_2$ that would be such that:

$$(x_1^0, x_2^1) \sim (x_1^1, x_2^0). \tag{3}$$

Clearly this will require the structure of X_2 to be adequately "rich" so as to find the level $x_2^1 \in X_2$ such that the reference preference interval on the first attribute $[x_1^0, x_1^1]$ is exactly matched by a preference interval of the same "length" on the second attribute $[x_2^0, x_2^1]$. Technically, this calls for a solvability assumption or, more restrictively, for the supposition that X_2 has a (topological) structure that is close to that of an interval of $\mathbb{R}$ and that $\succsim$ is "somehow" continuous.

If such a level x_2^1 can be found, model (2) implies:

$$\begin{aligned} &v_1(x_1^0) + v_2(x_2^1) = v_1(x_1^1) + v_2(x_2^0) \text{ so that} \\ &v_2(x_2^1) - v_2(x_2^0) = v_1(x_1^1) - v_1(x_1^0). \end{aligned} \tag{4}$$

Let us fix the origin of measurement letting:

$$v_1(x_1^0) = v_2(x_2^0) = 0,$$

and our unit of measurement letting:

$$v_1(x_1^1) = 1 \text{ so that } v_1(x_1^1) - v_1(x_1^0) = 1.$$

Using (4), we therefore obtain $v_2(x_2^1) = 1$. We have therefore found an interval between levels on the second attribute ($[x_2^0, x_2^1]$) that exactly matches our reference interval on the first attribute ($[x_1^0, x_1^1]$). We may proceed to build our standard sequence on the second attribute (see Figure 4) asking for levels $x_2^2, x_2^3, \ldots$ such that:

$$\begin{gathered} (x_1^0, x_2^2) \sim (x_1^1, x_2^1), \\ (x_1^0, x_2^3) \sim (x_1^1, x_2^2), \\ \ldots \\ (x_1^0, x_2^k) \sim (x_1^1, x_2^{k-1}). \end{gathered}$$

As above, using (2) leads to:

$$\begin{aligned} v_2(x_2^2) - v_2(x_2^1) &= v_1(x_1^1) - v_1(x_1^0), \\ v_2(x_2^3) - v_2(x_2^2) &= v_1(x_1^1) - v_1(x_1^0), \\ &\ldots \\ v_2(x_2^k) - v_2(x_2^{k-1}) &= v_1(x_1^1) - v_1(x_1^0), \end{aligned}$$

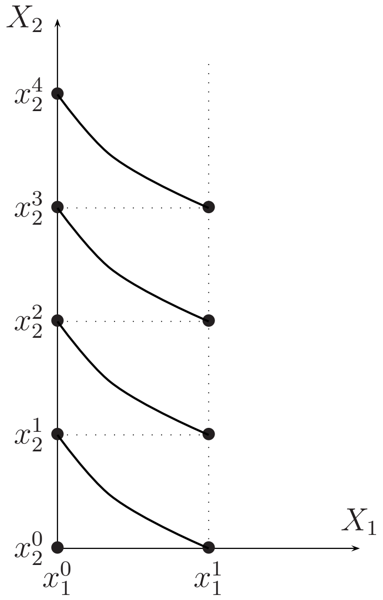


Figure 4: Building a standard sequence on X_2.

so that:

$$v_2(x_2^2) = 2, v_2(x_2^3) = 3, \ldots, v_2(x_2^k) = k.$$

This process of building a standard sequence of the second attribute therefore leads to defining v_2 on a number of, carefully, selected elements of X_2.

Remember the standard sequence that we built for length in section 1.2. An implicit hypothesis was that the length of any rod could be exceeded by the length of a composite object obtained by concatenating a sufficient number of perfect copies of a standard rod. Such an hypothesis is called “Archimedean” since it mimics the property of the real numbers saying that for any positive real numbers x, y it is true that $nx > y$ for some integer n, i.e. y, no matter how large, may always be exceeded by taking any x, no matter how small, and adding it with itself and repeating the operation a sufficient number of times. Clearly, we will need a similar hypothesis here. Failing it, there might exist a level $y_2 \in X_2$ that will never be “reached” by our standard sequence, i.e. such that $y_2 \succ_2 x_2^k$, for $k = 1, 2, \ldots$. For measurement models in which this Archimedean condition is omitted, see [155, 193].

Remark 16

At this point a good exercise for the reader is to figure out how we may extend the standard sequence to cover levels of X_2 that are “below” the reference level x_2^0. This should not be difficult. •

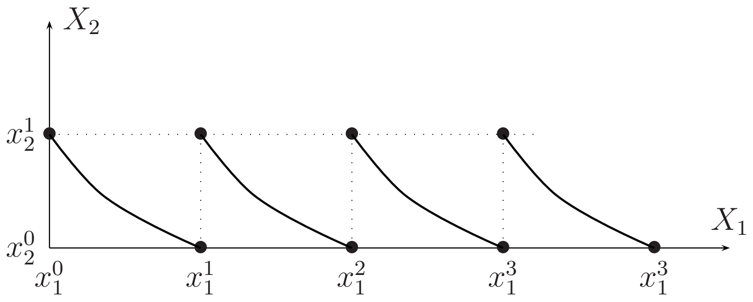


Figure 5: Building a standard sequence on X_1.

Now that a standard sequence is built on the second attribute, we may use any part of it to build a standard sequence on the first attribute. This will require finding levels $x_1^2, x_1^3, \ldots \in X_1$ such that (see Figure 5):

$$\begin{gathered}(x_1^2, x_2^0) \sim (x_1^1, x_2^1),\\ (x_1^3, x_2^0) \sim (x_1^2, x_2^1),\\ \ldots\\ (x_1^k, x_2^0) \sim (x_1^{k-1}, x_2^1).\end{gathered}$$

Using (2) leads to:

$$\begin{aligned}v_1(x_1^2) - v_1(x_1^1) &= v_2(x_2^1) - v_2(x_2^0),\\ v_1(x_1^3) - v_1(x_1^2) &= v_2(x_2^1) - v_2(x_2^0),\\ &\ldots\\ v_1(x_1^k) - v_1(x_1^{k-1}) &= v_2(x_2^1) - v_2(x_2^0),\end{aligned}$$

so that:

$$v_1(x_1^2) = 2, v_1(x_1^3) = 3, \ldots, v_1(x_1^k) = k.$$

As was the case for the second attribute, the construction of such a sequence will require the structure of X_1 to be adequately rich, which calls for a solvability assumption. An Archimedean condition will also be needed in order to be sure that all levels of X_1 can be reached by the sequence.

We have defined a "grid" in X (see Figure 6) and we have $v_1(x_1^k) = k$ and $v_2(x_2^k) = k$ for all elements of this grid. Intuitively such numerical assignments seem to define an adequate additive value function on the grid. We have to prove that this intuition is correct. Let us first verify that, for all integers $\alpha, \beta, \gamma, \delta$:

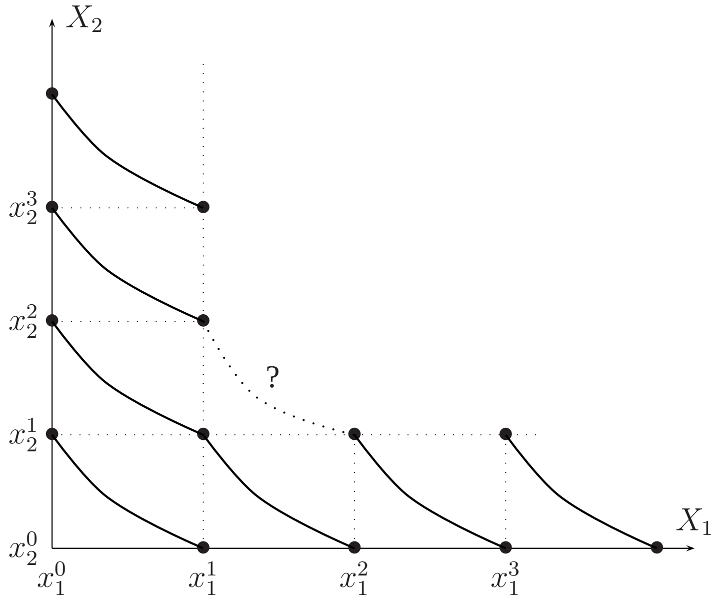


Figure 6: The grid.

$$\alpha + \beta = \gamma + \delta = \epsilon \Rightarrow (x_1^{\alpha}, x_2^{\beta}) \sim (x_1^{\gamma}, x_2^{\delta}). \tag{5}$$

When $\epsilon = 1$, (5) holds by construction because we have: $(x_1^0, x_2^1) \sim (x_1^1, x_2^0)$. When $\epsilon = 2$, we know that $(x_1^0, x_2^2) \sim (x_1^1, x_2^1)$ and $(x_1^2, x_2^0) \sim (x_1^1, x_2^1)$ and the claim is proved using the transitivity of $\sim$.

Consider the $\epsilon = 3$ case. We have $(x_1^0, x_2^3) \sim (x_1^1, x_2^2)$ and $(x_1^0, x_2^3) \sim (x_1^1, x_2^2)$. It remains to be shown that $(x_1^2, x_2^1) \sim (x_1^1, x_2^2)$ (see the dotted arc in Figure 6). This does not seem to follow from the previous conditions that we more or less explicitly used: transitivity, independence, “richness”, Archimedean. Indeed, it does not. Hence, we have to suppose that: $(x_1^2, x_2^0) \sim (x_1^0, x_2^2)$ and $(x_1^0, x_2^1) \sim (x_1^1, x_2^0)$ imply $(x_1^2, x_2^1) \sim (x_1^1, x_2^2)$. This condition, called the Thomsen condition, is clearly necessary for (2). The above reasoning easily extends to all points on the grid, using weak ordering, independence and the Thomsen condition. Hence, (5) holds on the grid.

It remains to show that:

$$\epsilon = \alpha + \beta > \epsilon' = \gamma + \delta \Rightarrow (x_1^{\alpha}, x_2^{\beta}) \succ (x_1^{\gamma}, x_2^{\delta}). \tag{6}$$

Using transitivity, it is sufficient to show that (6) holds when $\epsilon = \epsilon' + 1$. By construction, we know that $(x_1^1, x_2^0) \succ (x_1^0, x_2^0)$. Using independence this implies that $(x_1^1, x_2^k) \succ (x_1^0, x_2^k)$. Using (5) we have $(x_1^1, x_2^k) \sim (x_1^{k+1}, x_2^0)$ and $(x_1^0, x_2^k) \sim (x_1^k, x_2^0)$. Therefore we have $(x_1^{k+1}, x_2^0) \succ (x_1^k, x_2^0)$, the desired conclusion.

Hence, we have built an additive value function of a suitably chosen grid (see Figure 7). The logic of the assessment procedure is then to assess more and more points

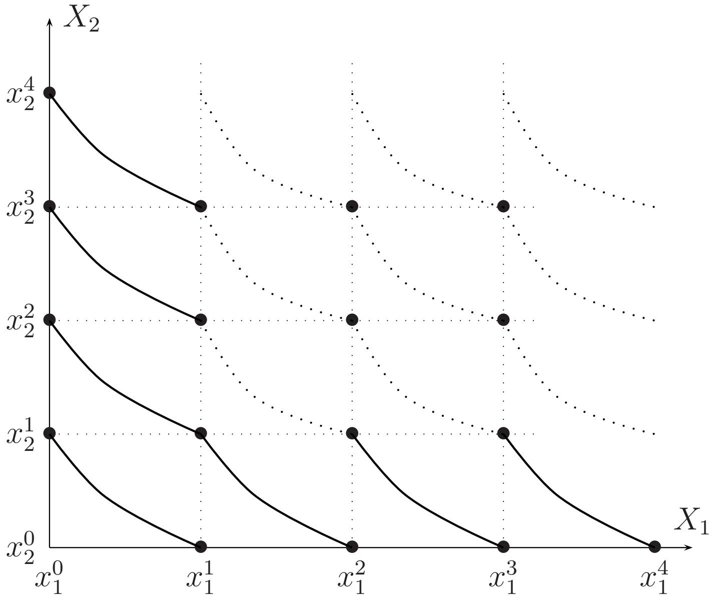


Figure 7: The entire grid.

somehow considering more finely grained standard sequences. The two techniques evoked for length may be used here depending on the underlying structure of X. Going to the limit then unambiguously defines the functions v_1 and v_2. Clearly such v_1 and v_2 are intimately related. Once we have chosen an arbitrary reference point (x_1^0, x_2^0) and a level x_1^1 defining the unit of measurement, the process just described entirely defines v_1 and v_2. It follows that the only possible transformations that can be applied to v_1 and v_2 is to multiply both by the same positive number α and to add to both a, possibly different, constant. This is usually summarized saying that v_1 and v_2 define interval scales with a common unit.

The above reasoning is a rough sketch of the proof of the existence of an additive value function when $n = 2$, as well as a sketch of how it could be assessed. Careful readers will want to refer to [58, 129, 211].

Remark 17

The measurement of length through standard sequences described above leads to a scale that is unique once the unit of measurement is chosen. At this point, a good exercise for the reader is to find an intuitive explanation to the fact that, when measuring the “length” of preference intervals, the origin of measurement becomes arbitrary. The analogy with the the measurement of duration on the one hand and dates, as given in a calendar, on the other hand should help. •

Remark 18

As was already the case with the even swaps technique, it is worth emphasizing that this assessment technique makes no use of the vague notion of the "importance" of the various attributes. The "importance" is captured here in the lengths of the preference intervals on the various attributes.

A common but critical mistake is to confuse the additive value function model (2) with a weighted average and to try to assess weights asking whether an attribute is "more important" than another. This makes no sense. •

3.1.2 The case of more than two attributes

The good news is that the process is exactly the same when there are more than two attributes. With one surprise: the Thomsen condition is no more needed to prove that the standard sequences defined on each attribute lead to an adequate value function on the grid. A heuristic explanation of this strange result is that, when $n = 2$, there is no difference between independence and weak independence. This is no more true when $n \geq 3$ and assuming independence is much stronger than just assuming weak independence.

3.2 Statement of results

We use below the "algebraic approach" [127, 129, 141]. A more restrictive approach using a topological structure on X is given in [44, 58, 211]. We formalize below the conditions informally introduced in the preceding section. The reader not interested in the precise statement of the results or, better, having already written down his own statement, may skip this section.

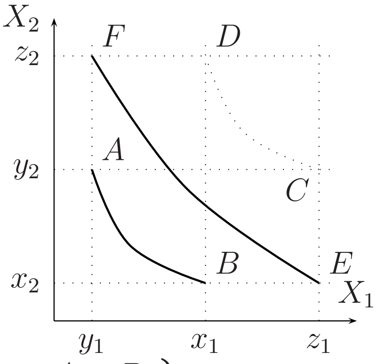
Definition 19 (Thomsen condition)

Let $\succsim$ be a binary relation on a set $X = X_1 \times X_2$. It is said to satisfy the Thomsen condition if

$$(x_1, x_2) \sim (y_1, y_2) \text{ and } (y_1, z_2) \sim (z_1, x_2) \Rightarrow (x_1, z_2) \sim (z_1, y_2),$$

for all $x_1, y_1, z_1 \in X_1$ and all $x_2, y_2, z_2 \in X_2$.

Figure 8 shows how the Thomsen condition uses two "indifference curves" (i.e. curves linking points that are indifferent) to place a constraint on a third one. This was needed above to prove that an additive value function existed on our grid. Remember that the Thomsen condition is only needed when $n = 2$; hence, we only stated it in this case.



$$\left.\begin{array}{l} A \sim B \\ E \sim F \end{array}\right\} \Rightarrow C \sim D$$

Figure 8: The Thomsen condition.

Definition 20 (Standard sequences)
A standard sequence on attribute $i \in N$ is a set $\{a_i^k : a_i^k \in X_i, k \in K\}$ where K is a set of consecutive integers (positive or negative, finite or infinite) such that there are $x_{-i}, y_{-i} \in X_{-i}$ satisfying $Not[\, x_{-i} \sim_{-i} y_{-i}\,]$ and $(a_i^k, x_{-i}) \sim (a_i^{k+1}, y_{-i})$, for all $k \in K$.

A standard sequence on attribute $i \in N$ is said to be *strictly bounded* if there are $b_i, c_i \in X_i$ such that $b_i \succ_i a_i^k \succ_i c_i$, for all $k \in K$. It is then clear that, when model (2) holds, any strictly bounded standard sequence must be finite.

Definition 21 (Archimedean)
For all $i \in N$, any strictly bounded standard sequence on $i \in N$ is finite.

The following condition rules out the case in which a standard sequence cannot be built because all levels are indifferent.

Definition 22 (Essentiality)
Let $\succsim$ be a binary relation on a set $X = X_1 \times X_2 \times \cdots \times X_n$. Attribute $i \in N$ is said to be essential if $(x_i, a_{-i}) \succ (y_i, a_{-i})$, for some $x_i, y_i \in X_i$ and some $a_{-i} \in X_{-i}$.

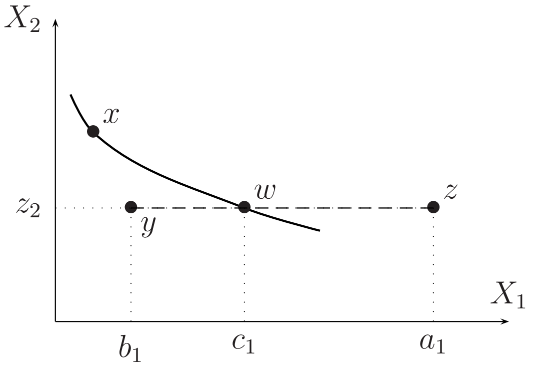
Definition 23 (Restricted Solvability)
Let $\succsim$ be a binary relation on a set $X = X_1 \times X_2 \times \cdots \times X_n$. Restricted solvability is said to hold with respect to attribute $i \in N$ if, for all $x \in X$, all $z_{-i} \in X_{-i}$ and all $a_i, b_i \in X_i$, $[(a_i, z_{-i}) \succsim x \succsim (b_i, z_{-i})] \Rightarrow [x \sim (c_i, z_{-i})$, for some $c_i \in X_i]$.

Remark 24
Restricted solvability is illustrated in Figure 9 in the case where $n = 2$. It says that, given any $x \in X$, if it is possible find two levels $a_i, b_i \in X_i$ such that when combined with a certain level $z_{-i} \in X_{-i}$ on the other attributes, (a_i, z_{-i}) is preferred to x and x is

preferred to (b_i, z_{-i}), it should be possible to find a level c_i, “in between” a_i and b_i, such that (c_i, z_{-i}) is exactly indifferent to x.

A much stronger hypothesis is unrestricted solvability asserting that for all $x \in X$ and all $z_{-i} \in X_{-i}$, $x \sim (c_i, z_{-i})$, for some $c_i \in X_i$. Its use leads however to much simpler proofs [58, 86].

It is easy to imagine situations in which restricted solvability might hold while unrestricted solvability would fail. Suppose, e.g. that a firm has to choose between several investment projects, two attributes being the Net Present Value (NPV) of the projects and their impact on the image of the firm in the public. Consider a project consisting in investing in the software market. It has a reasonable NPV and no adverse consequences on the image of the firm. Consider another project that could have dramatic consequences on the image of the firm, because it leads to investing the market of cocaine. Unrestricted solvability would require that by sufficiently increasing the NPV of the second project it would become indifferent to the more standard project of investing in the software market. This is not required by restricted solvability. •



$$\left.\begin{array}{l} z \succ x \\ x \succ y \end{array}\right\} \Rightarrow \text{ there is a } w \text{ such that } x \sim w$$

Figure 9: Restricted Solvability on X_1.

We are now in position to state the central results concerning model (2). Proofs may be found in [129, 213].

Theorem 25 (Additive value function when $n = 2$)
Let $\succsim$ be a binary relation on a set $X = X_1 \times X_2$. If restricted solvability holds on all attributes and each attribute is essential then $\succsim$ has a representation in model (2) *if and only if $\succsim$ is an independent weak order satisfying the Thomsen and the Archimedean conditions*

Furthermore in this representation, v_1 and v_2 are interval scales with a common unit, i.e. if v_1, v_2 and w_1, w_2 are two pairs of functions satisfying (2)*, there are real numbers α, β_1, β_2 with $\alpha > 0$ such that, for all $x_1 \in X_1$ and all $x_2 \in X_2$*

$$v_1(x_1) = \alpha w_1(x_1) + \beta_1 \textit{ and } v_2(x_2) = \alpha w_2(x_2) + \beta_2.$$

When $n \geq 3$ and at least three attributes are essential, the above result simplifies in that the Thomsen condition can now be omitted.

Theorem 26 (Additive value function when $n \geq 3$)
Let $\succsim$ be a binary relation on a set $X = X_1 \times X_2 \times \ldots \times X_n$ with $n \geq 3$. If restricted solvability holds on all attributes and at least 3 attributes are essential then $\succsim$ has a representation in model (2) *if and only if $\succsim$ is an independent weak order satisfying the Archimedean condition.*

Furthermore in this this representation v_1, $v_2, \ldots, v_n$ are interval scales with a common unit.

Remark 27
As mentioned in introduction, the additive value model is central to several fields in decision theory. It is therefore not surprising that much energy has been devoted to analyze variants and refinements of the above results. Among the most significant ones, let us mention:

- the study of cases in which solvability holds only on some or none of the attributes [75, 85, 86, 87, 88, 112, 113, 154],
- the study of the relation between the “algebraic approach” introduced above and the topological one used in [44], see e.g. [115, 124, 211, 213].

The above results are only valid when X is the entire Cartesian product of the sets X_i. Results in which X is a subset of the whole Cartesian product $X_1 \times X_2 \times \ldots \times X_n$ are not easy to obtain, see [37, 181] (the situation is “easier” in the special case of homogeneous product sets, see [214, 215]). •

3.3 Implementation: Standard sequences and beyond

We have already shown above how additive value functions can be assessed using the standard sequence technique. It is worth recalling here some of the characteristics of this assessment procedure:

- It requires the set X_i to be *rich* so that it is possible to find a preference interval on X_i that will exactly match a preference interval on another attribute. This excludes using such an assessment procedure when some of the sets X_i are discrete.

- It relies on *indifference* judgements which, a priori, are less firmly established than preference judgements.

- It relies on judgements concerning fictitious alternatives which, a priori, are harder to conceive than judgements concerning real alternatives.

- The various assessments are thoroughly intertwined and, e.g., an imprecision on the assessment of x_2^1, i.e. the endpoint of the first interval in the standard sequence on X_2 (see Figure 4) will propagate to many assessed values,

- The assessment of tradeoffs may be plagued with cognitive biases, see [46, 197].

The assessment procedure based on standard sequences is therefore rather demanding; this should be no surprise given the proximity between this form of measurement and extensive measurement illustrated above on the case of length. Hence, the assessment procedure based on standard sequences seems to be seldom used in the practice of decision analysis [121, 209]. The literature on the experimental assessment of additive value functions, see e.g. [197, 208, 216], suggests that this assessment is a difficult task that may be affected by several cognitive biases.

Many other simplified assessment procedures have been proposed that are less firmly grounded in theory. In many of them, the assessment of the partial value functions v_i relies on *direct* comparison of preference differences without recourse to an interval on another attribute used as a “meter stick”. We refer to [50, 209] for a theoretical analysis of these techniques.

These procedures include:

- *direct rating* techniques in which values of v_i are directly assessed with reference to two arbitrarily chosen points [52, 53],

- procedures based on *bisection*, the decision-maker being asked to assess a point that is “half way” in terms of preference two reference points [209],

- procedures trying to build *standard sequences* on each attribute in terms of “preference differences” [129, ch. 4].

An excellent overview of these techniques may be found in [209].

4 The additive value model in the "finite" case

4.1 Outline of theory

In this section, we suppose that $\succsim$ is a binary relation on a finite set $X \subseteq X_1 \times X_2 \times \cdots \times X_n$ (contrary to the preceding section, dealing with subsets of product sets will raise no difficulty here). The finiteness hypothesis clearly invalidates the standard sequence mechanism used till now. On each attribute there will only be finitely many "preference intervals" and exact matches between preference intervals will only happen exceptionally, see [212].

Clearly, independence remains a necessary condition for model (2) as before. Given the absence of structure of the set X, it is unlikely that this condition is sufficient to ensure (2). The following example shows that this intuition is indeed correct.

Example 28

Let $X = X_1 \times X_2$ with $X_1 = \{a, b, c\}$ and $X_2 = \{d, e, f\}$. Consider the weak order on X such that, abusing notation in an obvious way,

$$ad \succ bd \succ ae \succ af \succ be \succ cd \succ ce \succ bf \succ cf.$$

It is easy to check that $\succsim$ is independent. Indeed, we may for instance check that:

$$ad \succ bd \text{ and } ae \succ be \text{ and } af \succ bf,$$
$$ad \succ ae \text{ and } bd \succ be \text{ and } cd \succ ce.$$

This relation cannot however be represented in model (2) since:

$$\begin{aligned}
af \succ be &\Rightarrow v_1(a) + v_2(f) > v_1(b) + v_2(e),\\
be \succ cd &\Rightarrow v_1(b) + v_2(e) > v_1(c) + v_2(d),\\
ce \succ bf &\Rightarrow v_1(c) + v_2(e) > v_1(b) + v_2(f),\\
bd \succ ae &\Rightarrow v_1(b) + v_2(d) > v_1(a) + v_2(e).
\end{aligned}$$

Summing the first two inequalities leads to:

$$v_1(a) + v_2(f) > v_1(c) + v_2(d).$$

Summing the last two inequalities leads to:

$$v_1(c) + v_2(d) > v_1(a) + v_2(f),$$

a contradiction.

Note that, since no indifference is involved, the Thomsen condition is trivially satisfied. Although it is clearly necessary for model (2), adding it to independence will therefore not solve the problem. ◇

The conditions allowing to build an additive value model in the finite case were investigated in [1, 2, 179]. Although the resulting conditions turn out to be complex, the underlying idea is quite simple. It amounts to finding conditions under which a system of linear inequalities has a solution.

Suppose that $x \succ y$. If model (2) holds, this implies that:

$$\sum_{i=1}^{n} v_i(x_i) > \sum_{i=1}^{n} v_i(y_i). \tag{7}$$

Similarly if $x \sim y$, we obtain:

$$\sum_{i=1}^{n} v_i(x_i) = \sum_{i=1}^{n} v_i(y_i). \tag{8}$$

The problem is then to find conditions on $\succsim$ such that the system of finitely many equalities and inequalities (7-8) has a solution. This is a classical problem in Linear Algebra [83].

Definition 29 (Relation E^m)
Let m be an integer ≥ 2. Let $x^1, x^2, \dots, x^m, y^1, y^2, \dots, y^m \in X$. We say that

$$(x^1, x^2, \dots, x^m) E^m (y^1, y^2, \dots, y^m)$$

if, for all $i \in N$, $(x_i^1, x_i^2, \dots, x_i^m)$ is a permutation of $(y_i^1, y_i^2, \dots, y_i^m)$.

Suppose that $(x^1, x^2, \dots, x^m) E^m (y^1, y^2, \dots, y^m)$ then model (2) implies that

$$\sum_{j=1}^{m} \sum_{i=1}^{n} v_i(x_i^j) = \sum_{j=1}^{m} \sum_{i=1}^{n} v_i(y_i^j).$$

Therefore if $x^j \succsim y^j$ for $j = 1, 2, \dots, m-1$, it cannot be true that $x^m \succ y^m$. This condition must hold for all $m = 2, 3, \dots$.

Definition 30 (Condition C^m)
Let m be an integer ≥ 2. We say that condition C^m holds if

$$[x^j \succsim y^j \text{ for } j = 1, 2, \dots, m-1] \Rightarrow Not[\, x^m \succ y^m \,]$$

for all $x^1, x^2, \dots, x^m, y^1, y^2, \dots, y^m \in X$ such that

$$(x^1, x^2, \dots, x^m) E^m (y^1, y^2, \dots, y^m).$$

Remark 31
It is not difficult to check that:

- $C^{m+1} \Rightarrow C^m$,
- $C^2 \Rightarrow \succsim$ is independent,
- $C^3 \Rightarrow \succsim$ is transitive. •

We already observed that C^m was implied by the existence of an additive representation. The main result for the finite case states that requiring that $\succsim$ is complete and that C^m holds for $m = 2, 3, \dots$ is also sufficient. Proofs can be found in [58, 129].

Theorem 32
Let $\succsim$ be a binary relation on a finite set $X \subseteq X_1 \times X_2 \times \dots \times X_n$. There are real-valued functions v_i on X_i such that (2) *holds if and only if $\succsim$ is complete and satisfies C^m for $m = 2, 3, \dots$.*

Remark 33
Contrary to the “rich” case considered in the preceding section, we have here necessary and sufficient conditions for the additive value model (2). However, it is important to notice that the above result uses a denumerable scheme of conditions. It is shown in [180] that this denumerable scheme cannot be truncated: for all $m \geq 2$, there is a relation $\succsim$ on a finite set X such that C^m holds but violating C^{m+1}. This is studied in more detail in [139, 201, 218]. Therefore, no finite scheme of axioms is sufficient to characterize model (2) for all finite sets X.

Given a finite set X *of given cardinality*, it is well-known that the denumerable scheme of condition can be truncated. The precise relation between the cardinality of X and the number of conditions needed raises difficult combinatorial questions that are studied in [77, 78]. •

Remark 34
It is clear that, if a relation $\succsim$ has a representation in model (2) with functions v_i, it also has a representation using functions $v'_i = \alpha v_i + \beta_i$ with $\alpha > 0$. Contrary to the rich case, the uniqueness of the functions v_i is more complex as shown by the following example.

Example 35
Let $X = X_1 \times X_2$ with $X_1 = \{a, b, c\}$ and $X_2 = \{d, e\}$. Consider the weak order on X such that, abusing notation in an obvious way,

$$ad \succ bd \succ ae \succ cd \succ be \succ ce.$$

This relation has a representation in model (2) with

$$v_1(a) = 3, v_1(b) = 1, v_1(c) = 0, v_2(d) = 3, v_2(e) = 0.5.$$

An equally valid representation would be given taking $v_1(b) = 2$. Clearly this new representation cannot be deduced from the original one applying a positive affine transformation. ◇

•

Remark 36

Theorem 32 has been extended to the case of an arbitrary set X in [113, 112], see also [75, 81]. The resulting conditions are however quite complex. This explains why we spent time on this "rich" case in the preceding section. •

Remark 37

The use of a denumerable scheme of conditions in theorem 32 does not facilitate the interpretation and the test of conditions. However it should be noticed that, on a given set X, the test of the C^m conditions amounts to finding if a system of finitely many linear inequalities has a solution. It is well-known that Linear Programming techniques are quite efficient for such a task. •

4.2 Implementation: LP-based assessment

We show how to use LP techniques in order to assess an additive value model (2), without supposing that the sets X_i are rich. For practical purposes, it is not restrictive to assume that we are only interested in assessing a model for a limited range on each X_i. We therefore assume that the sets X_i are bounded so that, using independence, there is a worst value x_{i*} and a most preferable value x_i^*. Using the uniqueness properties of model (2), we may always suppose, after an appropriate normalization, that:

$$v_1(x_{1*}) = v_2(x_{2*}) = \ldots = v_n(x_{n*}) = 0 \text{ and} \tag{9}$$

$$\sum_{i=1}^{n} v_i(x_i^*) = 1. \tag{10}$$

Two main cases arise (see Figures 10 and 11):

- attribute $i \in N$ is discrete so that the evaluation of any conceivable alternative on this attribute belongs to a finite set. We suppose that $X_i = \{x_{i*}, x_i^1, x_i^2, \ldots, x_i^{r_i}, x_i^*\}$. We therefore have to assess $r_i + 1$ values of v_i,
- the attribute $i \in N$ has an underlying continuous structure. It is hardly restrictive in practice to suppose that $X_i \subset \mathbb{R}$, so that the evaluation of an alternative on this attribute may take any value between x_{i*} and x_i^*. In this case, we may opt for the assessment of a piecewise linear approximation of v_i partitioning the set X_i in $r_i + 1$ intervals and supposing that v_i is linear on each of these intervals. Note that the approximation of v_i can be made more precise simply by increasing the number of these intervals.

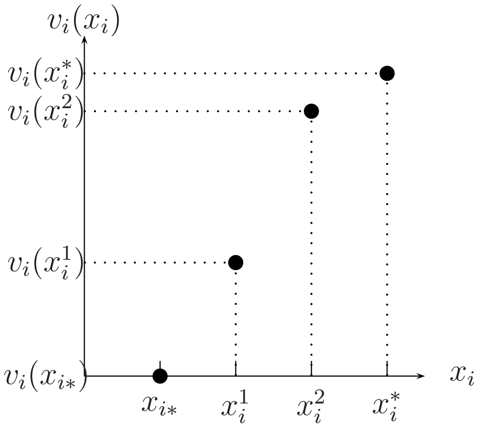


Figure 10: Value function when X_i is discrete.

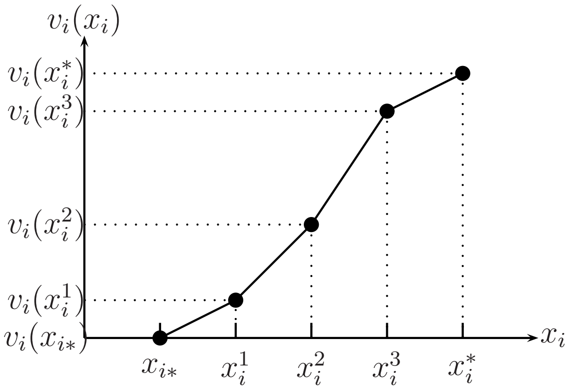


Figure 11: Value function when X_i is continuous.

With these conventions, the assessment of the model (2) amounts to giving a value to $\sum_{i=1}^{n}(r_i+1)$ unknowns. Clearly any judgment of preference linking x and y translate into a *linear inequality* between these unknowns. Similarly any judgment of indifference linking x and y translate into a *linear equality*. Linear Programming (LP) offers a powerful tool for testing whether such a system has solutions. Therefore, an assessment procedure can be conceived on the following basis:

- obtain judgments in terms of preference or indifference linking several alternatives in X,
- convert these judgments into linear (in)equalities,
- test, using LP, whether this system has a solution.

If the system has no solution then one may either propose a solution that will be “as close as possible” from the information obtained, e.g. violating the minimum number of (in)equalities or suggest the reconsideration of certain judgements. If the system has a solution, one may explore the set of all solutions to this system since they are all candidates for the establishment of model (2). These various techniques depend on:

- the choice of the alternatives in X that are compared: they may be real or fictitious, they may differ on a different number of attributes,
- the way to deal with the inconsistency of the system and to eventually propose some judgments to be reconsidered,
- the way to explore the set of solutions of the system and to use this set as the basis for deriving a prescription.

Linear programming offers of simple and versatile technique to assess additive value functions. All restrictions generating linear constraints of the coefficient of the value function can easily be accommodated. This idea has been often exploited, see [16]. We present below two techniques using it. It should be noticed that rather different techniques have been proposed in the literature on Marketing [35, 103, 104, 114, 132].

4.2.1 UTA [111]

UTA (“UTilité Additive”, i.e. additive utility in French) is one of the oldest techniques belonging to this family. It is supposed in UTA that there is a subset $Ref \subset X$ of reference alternatives that the decision-maker knows well either because he/she has experienced them or because they have received particular attention. The technique amounts to asking the DM to provide a weak order on Ref. Each preference or indifference relation contained in this weak order is then translated into a linear constraint:

- $x \sim y$ gives an equality $v(x) - v(y) = 0$ and
- $x \succ y$ gives an inequality $v(x) - v(y) > 0$,

where $v(x)$ and $v(y)$ can be expressed as a linear combination of the unknowns as remarked earlier. Strict inequalities are then translated into large inequalities as is usual in Linear Programming, i.e. $v(x) - v(y) > 0$ becomes $v(x) - v(y) \geq \epsilon$ where $\epsilon > 0$ is a very small positive number that should be chosen according to the precision of the arithmetics used by the LP package.

The test of the existence of a solution to the system of linear constraints is done via standard Goal Programming techniques [36] adding appropriate deviation variables. In

UTA, each equation $v(x)-v(y)=0$ is translated into an equation $v(x)-v(y)+\sigma_x^+-\sigma_x^-+\sigma_y^+-\sigma_y^-=0$, where $\sigma_x^+, \sigma_x^-, \sigma_y^+$ and σ_y^- are nonnegative deviation variables. Similarly each inequality $v(x)-v(y)\geq\epsilon$ is written as $v(x)-v(y)+\sigma_x^+-\sigma_x^-+\sigma_y^+-\sigma_y^-\geq\epsilon$. It is clear that there will exist a solution to the original system of linear constraints if there is a solution of the LP in which all deviation variables are zero. This can easily be tested using the objective function

$$\text{Minimize } Z=\sum_{x\in Ref}\sigma_x^+ + \sigma_x^- \tag{11}$$

Two cases arise. If the optimal value of Z is 0, there is an additive value function that represents the preference information. It should be observed that, except in exceptional cases (e.g. if the preference information collected is identical to the preference information collected with the standard sequence technique), there are infinitely many such additive value functions (that are not related via a simple change of origin and of unit, since we already fixed them through normalization (9-10)). The one given as the "optimal" one by the LP does not have a special status since it is highly dependent upon the arbitrary choice of the objective function; instead of minimizing the sum of the deviation variables, we could have as well, and still preserving linearity, minimized the largest of these variables. The whole polyhedron of feasible solutions of the original (in)equalities corresponds to adequate additive value functions: we have a whole set $\mathcal{V}$ of additive value functions representing the information collected on the set of reference alternatives Ref.

The size of $\mathcal{V}$ is clearly dependent upon the choice of the alternatives in Ref. Using standard techniques in LP, several functions in $\mathcal{V}$ may be obtained, e.g. the ones maximizing or minimizing, within $\mathcal{V}$, $v_i(x_i^*)$ for each attribute [111]. It is often interesting to present them to the decision-maker in the pictorial form of Figures 10 and 11.

If the optimal value of Z is strictly greater than 0, there is no additive value function representing the preference information available. The solution given as optimal (note that it is not guaranteed that this solution leads to the minimum possible number of violations w.r.t. the information provided—this would require solving an integer linear programme) is, in general, highly dependent upon the choice of the objective function.

This absence of solution to the system might be due to several factors:

- the piecewise linear approximation of the v_i for the "continuous" attributes may be too rough. It is easy to test whether an increase in the number of linear pieces on some of these attributes may lead to a nonempty set of additive value functions.
- the information provided by the decision-maker may be of poor quality. It might then be interesting to present to the decision-maker one additive value function (e.g. one may present an average function after some post-optimality analysis) in the pictorial form of Figures 10 and 11 and to let him react to this information either by

modifying his/her initial judgments or even by letting him/her react directly on the shape of the value functions. This is the solution implemented in the well-known PREFCALC system [109].

- the preference provided by the decision-maker might be inconsistent with the conditions implied by an additive value function. The system should then help locate these inconsistencies and allow the DM to think about them. Alternatively, since many alternative attribute descriptions are possible, it may be worth investigating whether a different definition of the various attributes may lead to a preference model consistent with model (2). Several examples of such analysis may be found in [119, 121, 209]

When the above techniques fail, the optimal solution of the LP, even if not compatible with the information provided, may still be considered as an adequate model. Again, since the objective function introduced above is somewhat arbitrary and it is recommended in [111] to perform a post-optimality analysis, e.g. considering additive value functions that are "close" to the optimal solution through the introduction of a linear constraint:

$$Z \leq Z^* + \delta,$$

where Z^* is the optimal value of the objective function of the original LP and δ is a "small" positive number. As above, the result of the analysis is a set $\mathcal{V}$ of additive value functions defined by a set of linear constraints. A representative sample of additive value functions within $\mathcal{V}$ may be obtained as above.

It should be noted that many possible variants of UTA can be conceived building on the following comments. They include:

- the addition of monotonicity properties of the v_i with respect to the underlying continuous attributes,
- the addition of constraints on the shape of the marginal value functions v_i, e.g. requiring them to be concave, convex or S-shaped,
- the addition of constraints linked to a possible indication of preference intensity for the elements of Ref given by the DM, e.g. the difference between x and y is larger than the difference between z and w.

For applications of UTA-like techniques, we refer to [38, 47, 48, 105, 110, 148, 185, 186, 187, 188, 189, 190, 192, 195, 196, 219, 221, 220, 223, 222]. Variants of the method are considered in [19, 20, 191].

4.2.2 MACBETH [12]

It is easy to see that (9) and (10) may equivalently be written as:

$$x \succsim y \Leftrightarrow \sum_{i=1}^{n} k_i u_i(x_i) \geq \sum_{i=1}^{n} k_i u_i(y_i), \tag{12}$$

where

$$u_1(x_{1*}) = u_2(x_{2*}) = \ldots u_n(x_{n*}) = 0, \tag{13}$$
$$u_1(x_1^*) = u_2(x_2^*) = \ldots u_n(x_n^*) = 1 \text{ and} \tag{14}$$
$$\sum_{i=1}^{n} k_i = 1. \tag{15}$$

With such an expression of an additive value function, it is tempting to break down the assessment into two distinct parts: a value function u_i is assessed on each attribute and, then, scaling constants k_i are assessed taking the shape of the value functions u_i as given. This is the path followed in MACBETH.

Remark 38

Again, note that we are speaking here of k_i as *scaling constants* and not as *weights*. As already mentioned weights that would reflect the "importance" of attributes are irrelevant to assess the additive value function model. Notice that, under (12-15) the ordering of the scaling constant k_i is dependent upon the choice of x_{i*} and x_i^*. Increasing the width of the interval $[x_{i*}, x_i^*]$ will lead to increasing the value of the scaling constant k_i. The value k_i has, therefore, nothing to do with the "importance" of attribute i. This point is unfortunately too often forgotten when using a weighted average of some numerical attributes. In the latter model, changing the units in which the attributes are measured should imply changing the "weights" accordingly. •

The assessment procedure of the u_i is conceived in such a way as to avoid comparing alternatives differing on more than one attribute. In view of what was said before concerning the standard sequence technique, this is clearly an advantage of the technique. But can it be done? The trick here is that MACBETH asks for judgments related to the difference between the desirability of alternatives and not only judgments in terms of preference or indifference. Partial value functions u_i are approximated in a similar way than in UTA: for discrete attributes, each point on the function is assessed, for continuous ones, a piecewise linear approximation is used.

MACBETH asks the DM to compare pairs of levels on each attribute. If no difference is felt between these levels, they receive an identical partial value level. If a difference is felt between x_i^k and x_i^r, MACBETH asks for a judgment qualifying the strength of this difference. The method and the associated software propose three different semantical categories:

Categories	Description
$C1$	weak
$C2$	strong
$C3$	extreme

with the possibility of using intermediate categories, i.e. between null and weak, weak and strong, strong and extreme (giving a total of six distinct categories). This information is then converted into linear inequations using the natural interpretation that if the "difference" between the levels x_i^k and x_i^r has been judged larger than the "difference" between $x_i^{k'}$ and $x_i^{r'}$ then it should follow that $u_i(x_i^k) - u_i(x_i^r) > u_i(x_i^{k'}) - u_i(x_i^{r'})$. Technically the six distinct categories are delimited by thresholds that are used in the establishment of the constraints of the LP. The software associated to MACBETH offers the possibility to compare all pairs of levels on each attribute for a total of $(r_i + 1)r_i/2$ comparisons. Using standard Goal Programming techniques, as in UTA, the test of the compatibility of a partial value function with this information is performed via the solution of a LP. If there is a partial value function compatible with the information, a "central" function is proposed to the DM who has the possibility to modify it. If not, the results of the LP are exploited in such a way to propose modifications of the information that would make it consistent.

The assessment of the scaling constant k_i is done using similar principles. The DM is asked to compare the following $(n + 2)$ alternatives by pairs:

$$\begin{gathered}(x_{1*}, x_{2*}, \dots, x_{n*}),\\ (x_1^*, x_{2*}, \dots, x_{n*}),\\ (x_{1*}, x_2^*, \dots, x_{n*}),\\ \dots\\ (x_{1*}, x_{2*}, \dots, x_n^*) \text{ and}\\ (x_1^*, x_2^*, \dots, x_n^*),\end{gathered}$$

placing each pair in a category of difference. This information immediately translates into a set of linear constraints on the k_i. These constraints are processed as before. It should be noticed that, once the partial value functions u_i are assessed, it is not necessary to use the levels x_{i*} and x_i^* to assess the k_i since they may well lead to alternatives that are too unrealistic. The authors of MACBETH suggest to replace x_{i*} by a "neutral" level which appears neither desirable nor undesirable and x_i^* by a "desirable" level that is judged satisfactory. Although this clearly impacts the quality of the dialogue with the DM, this has no consequence on the underlying technique used to process information.

We refer to [6, 7, 8, 9, 10, 11] for applications of the MACBETH technique.

5 Extensions

The additive value model (2) is the central model for the application of conjoint measurement techniques to decision analysis. In this section, we consider various extensions to this model.

5.1 Transitive Decomposable models

The transitive decomposable model has been introduced in [129] as a natural generalization of model (2). It amounts to replacing the addition operation by a general function that is increasing in each of its arguments.

Definition 39 (Transitive decomposable model)
Let $\succsim$ be a binary relation on a set $X = \prod_{i=1}^{n} X_i$. The transitive decomposable model holds if, for all $i \in N$, there is a real-valued function v_i on X_i and a real-valued function g on $\prod_{i=1}^{n} v_i(X_i)$ that is increasing in all its arguments such that:

$$x \succsim y \Leftrightarrow g(v_1(x_1), \ldots, v_n(x_n)) \geq g(v_1(y_1), \ldots, v_n(y_n)), \tag{16}$$

for all $x, y \in X$.

An interesting point with this model is that it admits an intuitively appealing simple characterization. The basic axiom for characterizing the above transitive decomposable model is weak independence, which is clearly implied by (16). The following theorem is proved in [129, ch. 7].

Theorem 40
A preference relation $\succsim$ on a finite or countably infinite set X has a representation in the transitive decomposable model iff $\succsim$ is a weakly independent weak order.

Remark 41
This result can be extended to sets of arbitrary cardinality adding a, necessary, condition implying that the weak order $\succsim$ has a numerical representation, see [42, 45]. •

The weak point of such a model is that the function g is left unspecified so that the model will be difficult to assess. Furthermore, the uniqueness results for v_i and g are clearly much less powerful than what we obtained with model (2), see [129, ch. 7]. Therefore, practical applications of this model generally imply specifying the type of function g, possibly by verifying further conditions on the preference relation that impose that g belongs to some parameterized family of functions, e.g. some polynomial function of the v_i. This is studied in detail in [129, ch. 7] and [14, 82, 139, 138, 156, 166, 202]. Since such models have, to the best of our knowledge, never been used in decision analysis, we do not analyze them further.

The structure of the decomposable model however suggests that assessment techniques for this model could well come from Artificial Intelligence with its "rule induction" machinery. Indeed the function g in model (16) may also be seen as a set of "rules". We refer to [97, 98, 100, 101] for a thorough study of the potentiality of such an approach.

Remark 42

A simple extension of the decomposable model consists in simply asking for a function g that would be nondecreasing in each of its arguments. The following result is proved in [30] (see also [100]) (it can easily be extended to cover the case of an arbitrary set X, adding a, necessary, condition implying that $\succsim$ has a numerical representation).

We say that $\succsim$ is weakly separable if, for all $i \in N$ and all $x_i, y_i \in X_i$, it is never true that $(x_i, z_{-i}) \succ (y_i, z_{-i})$ and $(y_i, w_{-i}) \succ (x_i, w_{-i})$, for some $z_{-i}, w_{-i} \in X_{-i}$. Clearly this is a weakening of weak independence since it tolerates to have at the same time $(x_i, z_{-i}) \succ (y_i, z_{-i})$ and $(x_i, w_{-i}) \sim (y_i, w_{-i})$.

Theorem 43

A preference relation $\succsim$ on a finite or countably infinite set X has a representation in the weak decomposable model:

$$x \succsim y \Leftrightarrow g(u_1(x_1), \ldots, u_n(x_n)) \geq g(u_1(y_1), \ldots, u_n(y_n))$$

with g nondecreasing in all its arguments iff $\succsim$ is a weakly separable weak order.

A recent trend of research has tried to characterize special functional forms for g in the weakly decomposable model, such as $\max$, $\min$ or some more complex forms. The main references include [26, 100, 102, 182, 194]. •

Remark 44

The use of "fuzzy integrals" as tools for aggregating criteria has recently attracted much attention [49, 90, 91, 93, 94, 95, 143, 145, 144, 146], the Choquet Integral and the Sugeno integral being among the most popular. It should be strongly emphasized that the very definition of these integrals requires to have at hand a weak order on $\cup_{i=1}^{n} X_i$, supposing w.l.o.g. that the sets X_i are disjoint. This is usually called a "commensurability hypothesis". Whereas this hypothesis is quite natural when dealing with an homogeneous Cartesian product, as in decision under uncertainty (see e.g. [211]), it is far less so in the area of multiple criteria decision making. A neat conjoint measurement analysis of such models and their associated assessment procedures is an open research question, see [92]. •

5.2 Intransitive indifference

Decomposable models form a large family of preferences though not large enough to encompass all cases that may be encountered when asking subjects to express preferences. A major restriction is that not all preferences may be assumed to be weak orders. The example of the sequence of cups of coffee, each differing from the previous one by an imperceptible quantity of sugar added [133], is famous; it leads to the notions of semiorder and interval order [4, 57, 66, 133, 161], in which indifference is not transitive, while strict preference is.

Ideally, taking intransitive indifference into account, we would want to arrive at a generalization of (2) in which:

$$\begin{gathered} x \sim y \Leftrightarrow |V(x) - V(y)| \leq \epsilon, \\ x \succ y \Leftrightarrow V(x) > V(y) + \epsilon, \end{gathered}$$

where $\epsilon \geq 0$ and $V(x) = \sum_{i=1}^{n} v_i(x_i)$.

In the finite case, it is not difficult to extend the conditions presented in section 4 to cover such a case. Indeed, we are still looking here for the solution to a system of linear constraints. Although this seems to have never been done, it would not be difficult to adapt the LP-based assessment techniques to this case.

On the contrary, extending the standard sequence technique of section 3 is a formidable challenge. Indeed, remember that these techniques crucially rest on indifference judgments which lead to the determination of "perfect copies" of a given preference interval. As soon as indifference is not supposed to be transitive, "perfect copies" are not so perfect and much trouble is expected. We refer to [84, 128, 134, 161, 198] for a study of these models.

Remark 45

Even if the analysis of such models proves difficult, it should be noted that the semi-ordered version of the additive value model may be interpreted as having a "built-in" sensitivity analysis via the introduction of the threshold ϵ. Therefore, in practice, we may usefully view ϵ not as a parameter to be assessed but as a simple trick to avoid undue discrimination, because of the imprecision inevitably involved in our assessment procedures, between close alternatives •

Remark 46

Clearly the above model can be generalized to cope with a possibly non-constant threshold. The literature on the subject remains minimal however, see [161]. •

5.3 Nontransitive preferences

Many authors [147, 203] have argued that the reasonableness of supposing that strict preference is transitive is not so strong when it comes to comparing objects evaluated on several attributes. As soon as it is supposed that subjects may use an “ordinal” strategy for comparing objects, examples inspired from the well-known Condorcet paradox [176, 183] show that intransitivities will be difficult to avoid. Indeed it is possible to observe predictable intransitivities of strict preference in carefully controlled experiments [203]. There may therefore be a descriptive interest to studying such models. When it comes to decision analysis, intransitive preferences are often dismissed on two grounds:

- on a practical level, it is not easy to build a recommendation on the basis of a binary relation in which $\succ$ would not be transitive. Indeed, social choice theorists, facing a similar problem, have devoted much effort to devising what could be called reasonable procedures to deal with such preferences [41, 62, 130, 131, 149, 158, 178]. This literature does not lead, as was expected, to the emergence of a single suitable procedure in all situations.
- on a more conceptual level, many others have questioned the very rationality of such preferences using some version of the famous “money pump” argument [137, 164].

P. C. Fishburn has forcefully argued [73] that these arguments might not be as decisive as they appear at first sight. Furthermore some MCDM techniques make use of such intransitive models, most notably the so-called outranking methods [25, 172, 204, 205]. Besides the intellectual challenge, there might therefore be a real interest in studying such models.

A. Tversky [203] was one of the first to propose such a model generalizing (2), known as the *additive difference model*, in which:

$$x \succsim y \Leftrightarrow \sum_{i=1}^{n} \Phi_i(u_i(x_i) - u_i(y_i)) \geq 0 \tag{17}$$

where Φ_i are increasing and odd functions.

It is clear that (17) allows for intransitive $\succsim$ but implies its completeness. Clearly, (17) implies that $\succsim$ is independent. This allows to unambiguously define marginal preferences $\succsim_i$. Although model (17) can accommodate intransitive $\succsim$, a consequence of the increasingness of the Φ_i is that the marginal preference relations $\succsim_i$ are weak orders. This, in particular, excludes the possibility of any perception threshold on each attribute which would lead to an intransitive indifference relation on each attribute. Imposing that Φ_i are nondecreasing instead of being increasing allows for such a possibility. This gives rise to what is called the “weak additive difference model” in [22].

As suggested in [22, 70, 69, 72, 206], the subtractivity requirement in (17) can be relaxed. This leads to *nontransitive additive* conjoint measurement models in which:

$$x \succsim y \Leftrightarrow \sum_{i=1}^{n} p_i(x_i, y_i) \geq 0 \tag{18}$$

where the p_i are real-valued functions on X_i^2 and may have several additional properties (e.g. $p_i(x_i, x_i) = 0$, for all $i \in \{1, 2, \ldots, n\}$ and all $x_i \in X_i$).

This model is an obvious generalization of the (weak) additive difference model. It allows for intransitive and incomplete preference relations $\succsim$ as well as for intransitive and incomplete marginal preferences $\succsim_i$. An interesting specialization of (18) obtains when p_i are required to be *skew symmetric* i.e. such that $p_i(x_i, y_i) = -p_i(y_i, x_i)$. This skew symmetric nontransitive additive conjoint measurement model implies that $\succsim$ is complete and independent.

An excellent overview of these nontransitive models is [73]. Several axiom systems have been proposed to characterize them. P. C. Fishburn gave [70, 69, 72] axioms for the skew symmetric version of (18) both in the finite and the infinite case. Necessary and sufficient conditions for a nonstandard version of (18) are presented in [76]. [206] gives axioms for (18) with $p_i(x_i, x_i) = 0$ when $n \geq 4$. [22] gives necessary and sufficient conditions for (18) with and without skew symmetry in the denumerable case when $n = 2$.

The additive difference model (17) was axiomatized in [74] in the infinite case when $n \geq 3$ and [22] gives necessary and sufficient conditions for the weak additive difference model in the finite case when $n = 2$. Related studies of nontransitive models include [39, 64, 136, 153]. The implications of these models for decision-making under uncertainty were explored in [71] (for a different path to nontransitive models for decision making under risk and/or uncertainty, see [65, 67]).

It should be noticed that even the weakest form of these models, i.e. (18) without skew symmetry, involves an addition operation. Therefore it is unsurprising that the axiomatic analysis of these models share some common features with the additive value function model (2). Indeed, except in the special case in which $n = 2$, this case relating more to ordinal than to conjoint measurement (see [72]), the various axiom systems that have been proposed involve either:

- a denumerable set of cancellation conditions in the finite case or,
- a finite number of cancellation conditions together with unnecessary structural assumptions in the general case (these structural assumptions generally allow us to obtain nice uniqueness results for (18): the functions p_i are unique up to the multiplication by a common positive constant).

A different path to the analysis of nontransitive conjoint measurement models has recently been proposed in [30, 29, 31]. In order to get a feeling for these various models, it is useful to consider the various strategies that are likely to be implemented when comparing objects differing on several dimensions [40, 151, 152, 175, 200, 203].

Consider two alternatives x and y evaluated on a family of n attributes so that $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$.

A first strategy that can be used in order to decide whether or not it can be said that "x is at least as good as y" consists in trying to measure the "worth" of each alternative on each attribute and then to combine these evaluations adequately. Giving up all idea of transitivity and completeness, this suggests a model in which:

$$x \succsim y \Leftrightarrow F(u_1(x_1), \ldots, u_n(x_n), u_1(y_1), \ldots, u_n(y_n)) \geq 0 \tag{19}$$

where u_i are real-valued functions on the X_i and F is a real-valued function on $\prod_{i=1}^{n} u_i(X_i)^2$. Additional properties on F, e.g. its nondecreasingness (resp. nonincreasingness) in its first (resp. last) n arguments, will give rise to a variety of models implementing this first strategy.

A second strategy relies on the idea of measuring "preference differences" separately on each attribute and then combining these (positive or negative) differences in order to know whether the aggregation of these differences leads to an advantage for x over y. More formally, this suggests a model in which:

$$x \succsim y \Leftrightarrow G(p_1(x_1, y_1), p_2(x_2, y_2), \ldots, p_n(x_n, y_n)) \geq 0 \tag{20}$$

where p_i are real-valued functions on X_i^2 and G is a real-valued function on $\prod_{i=1}^{n} p_i(X_i^2)$. Additional properties on G (e.g. its oddness or its nondecreasingness in each of its arguments) or on p_i (e.g. $p_i(x_i, x_i) = 0$ or $p_i(x_i, y_i) = -p_i(y_i, x_i)$) will give rise to a variety of models in line with the above strategy.

Of course these two strategies are not incompatible and one may well consider using the "worth" of each alternative on each attribute to measure "preference differences". This suggests a model in which:

$$x \succsim y \Leftrightarrow H(\phi_1(u_1(x_1), u_1(y_1)), \ldots, \phi_n(u_n(x_n), u_n(y_n))) \geq 0 \tag{21}$$

where u_i are real-valued functions on X_i, ϕ_i are real-valued functions on $u_i(X_i)^2$ and H is a real-valued function on $\prod_{i=1}^{n} \phi_i(u_i(X_i)^2)$.

The use of general functional forms, instead of additive ones, greatly facilitate the axiomatic analysis of these models. It mainly relies on the study of various kinds of *traces* induced by the preference relation on coordinates and does not require a detailed analysis of tradeoffs between attributes.

The price to pay for such an extension of the scope of conjoint measurement is that the number of parameters that would be needed to assess such models is quite high. Furthermore, none of them is likely to possess any remarkable uniqueness properties. Therefore, although proofs are constructive, these results will not give direct hints on how to devise assessment procedures. The general idea here is to use numerical representations as guidelines to understand the consequences of a limited number of cancellation conditions, without imposing any transitivity or completeness requirement on the preference relation and any structural assumptions on the set of objects. Such models have proved useful to:

- understand the ordinal character of some aggregation models proposed in the literature [170, 172], known as the "outranking methods" as shown in [28],
- understand the links between aggregation models aiming at enriching a dominance relation and more traditional conjoint measurement approaches [30],
- to include in a classical conjoint measurement framework, noncompensatory preferences in the sense of [22, 33, 55, 60, 61] as shown in [28, 32, 99].

Acknowledgments

We thank two anonymous referees for their very helpful comments and suggestions. Part of this work was accomplished while Denis Bouyssou was visiting the *Service de Mathématique de la Gestion* at the *Université Libre de Bruxelles* (Brussels, Belgium). The warm hospitality of the *Service de Mathématique de la Gestion*, the support of the Belgian *Fonds National de la Recherche Scientistic* and the Brussels-Capital Region through a "Research in Brussels" action grant are gratefully acknowledged. Many Figures in this text were inspired by the ones drawn by Christophe Gonzales in [86].

References

[1] E. W. Adams. Elements of a theory of inexact measurement. *Philosophy of Science*, 32:205–228, 1965.

[2] E. W. Adams and R. F. Fagot. A model of riskless choice. *Behavioral Science*, 4:1–10, 1959.

[3] L. Adelman, P. J. Sticha, and M. L. Donnell. An experimental investigation of the relative effectiveness of two techniques for structuring multiattributed hierarchies. *Organizational Behavior and Human Decision Processes*, 37:188–196, 1986.

[4] F. Aleskerov and B. Monjardet. *Utility Maximization, Choice and Preference*. Springer Verlag, Heidelberg, 2002.

[5] A. B. Atkinson. On the measurement of inequality. *Journal of Economic Theory*, 2:244–263, 1970.

[6] C. A. Bana e Costa. The use of multi-criteria decision analysis to support the search for less conflicting policy options in a multi-actor context: Case study. *Journal of Multi-Criteria Decision Analysis*, 10(2):111–125, 2001.

[7] C. A. Bana e Costa, É. C. Corrêa, J.-M. De Corte, and J.-C. Vansnick. Facilitating bid evaluation in public call for tenders: A socio-technical approach. *Omega*, 30(3):227–242, April 2002.

[8] C. A. Bana e Costa, M. L. Costa-Lobo, I. A. J. Ramos, and J.-C. Vansnick. Multicriteria approach for strategic town planning: The case of Barcelos. In D. Bouyssou, É. Jacquet-Lagrèze, P. Perny, R. Słowiński, D. Vanderpooten, and Ph. Vincke, editors, *Aiding Decisions with Multiple Criteria: Essays in Honour of Bernard Roy*, pages 429–456. Kluwer, Dordrecht, 2002.

[9] C. A. Bana e Costa, L. Ensslin, É. C. Corrêa, and J.-C. Vansnick. Decision support systems in action: Integrated application in a multicriteria decission aid process. *European Journal of Operational Research*, 113(2):315–335, March 1999.

[10] C. A. Bana e Costa, F. Nunes da Silva, and J.-C. Vansnick. Conflict dissolution in the public sector: A case-study. *European Journal of Operational Research*, 130(2):388–401, April 2001.

[11] C. A. Bana e Costa and R. C. Oliveira. Assigning priorities for maintenance and repair and refurbishment in managing a municipal housing stock. *European Journal of Operational Research*, 138:380–91, 2002.

[12] C. A. Bana e Costa and J.-C. Vansnick. MACBETH – An interactive path towards the construction of cardinal value functions. *International Transactions in Operational Research*, 1:489–500, 1994.

[13] A. F. Beardon, J. C. Candeal, G. Herden, E. Induráin, and G. B. Mehta. The non-existence of a utility function and the structure of non-representatble preference relations. *Journal of Mathematical Economics*, 37:17–38, 2002.

[14] D. E. Bell. Multilinear representations for ordinal utility functions. *Journal of Mathematical Psychology*, 31:44–59, 1987.

[15] V. Belton, F. Ackermann, and I. Shepherd. Integrated support from problem structuring through alternative evaluation using COPE and V•I•S•A. *Journal of Multi-Criteria Decision Analysis*, 6:115–130, 1997.

[16] V. Belton and T. Stewart. *Multiple Criteria Decision Analysis: An Integrated Approach*. Kluwer, Dordrecht, 2001.

[17] E. Ben-Porath and I. Gilboa. Linear measures, the Gini index and the income-equality tradeoff. *Journal of Economic Theory*, 64:443–467, 1994.

[18] E. Ben-Porath, I. Gilboa, and D. Schmeidler. On the measurement of inequality under uncertainty. *Journal of Economic Theory*, 75:194–204, 1997.

[19] M. Beuthe and G. Scannella. Applications comparées des méthodes d'analyse multicritère UTA. *RAIRO Recherche Opérationnelle / Operations Research*, 30(3):293–315, 1996.

[20] M. Beuthe and G. Scannella. Comparative analysis of UTA multicriteria methods. *European Journal of Operational Research*, 130(2):246–262, 2001.

[21] C. Blackorby, D. Primont, and R. Russell. *Duality, separability, and functional structure: Theory and economic applications*. North-Holland, New York, 1978.

[22] D. Bouyssou. Some remarks on the notion of compensation in MCDM. *European Journal of Operational Research*, 26:150–160, 1986.

[23] D. Bouyssou. Modelling inaccurate determination, uncertainty, imprecision using multiple criteria. In A. G. Lockett and G. Islei, editors, *Improving Decision Making in Organisations*, pages 78–87. Springer Verlag, Heidelberg, 1989.

[24] D. Bouyssou. Builing criteria: A prerequisite for MCDA. In C. A. Bana e Costa, editor, *Readings in Multiple Criteria Decision Aid*, pages 58–81, Berlin, 1990. Springer Verlag.

[25] D. Bouyssou. Outranking methods. In C. A. Floudas and P. M. Pardalos, editors, *Encyclopedia of optimization*, volume 4, pages 249–255. Kluwer, 2001.

[26] D. Bouyssou, S. Greco, B. Matarazzo, M. Pirlot, and R. Słowiński. Characterization of 'max', 'min' and 'order statistics' multicriteria aggregation functions. Communication to *IFORS'2002*, 8 – 12 July, 2002, Edinburgh, U.K., July 2002.

[27] D. Bouyssou, Th. Marchant, M. Pirlot, P. Perny, A. Tsoukiàs, and Ph. Vincke. *Evaluation and Decision models: A critical perspective*. Kluwer, Dordrecht, 2000.

[28] D. Bouyssou and M. Pirlot. A characterization of strict concordance relations. In D. Bouyssou, É. Jacquet-Lagrèze, P. Perny, R. Słowiński, D. Vanderpooten, and Ph. Vincke, editors, *Aiding Decisions with Multiple Criteria: Essays in Honour of Bernard Roy*, pages 121–145. Kluwer, Dordrecht, 2002.

[29] D. Bouyssou and M. Pirlot. Nontransitive decomposable conjoint measurement. *Journal of Mathematical Psychology*, 46:677–703, 2002.

[30] D. Bouyssou and M. Pirlot. Preferences for multiattributed alternatives: Traces, dominance, and numerical representations. Working Paper, LAMSADE, Université Paris-Dauphine, submitted, 2002.

[31] D. Bouyssou and M. Pirlot. 'Additive difference' models without additivity and subtractivity. Working Paper, LAMSADE, Université Paris-Dauphine, submitted, 2003.

[32] D. Bouyssou and M. Pirlot. A characterization of concordance relations. Working Paper, LAMSADE, Université Paris-Dauphine, 2003.

[33] D. Bouyssou and J.-C. Vansnick. Noncompensatory and generalized noncompensatory preference structures. *Theory and Decision*, 21:251–266, 1986.

[34] D. S. Briges and G. B. Mehta. *Representations of preference orderings*. Springer Verlag, Berlin, 1995.

[35] J. D. Carroll and P. E. Green. Psychometric methods in marketing research. Part 1. Conjoint analysis. *Journal of Marketing Research*, 32:385–391, 1995.

[36] A. Charnes and W. W. Cooper. *Management Models and Industrial Applications of Linear Programming*. Wiley, New York, 1961.

[37] A. Chateauneuf and P. P. Wakker. From local to global additive representation. *Journal of Mathematical Economics*, 22(6):523–545, 1993.

[38] J.-C. Cosset, Y. Siskos, and C. Zopounidis. The evaluation of country risk: A decision support approach. *Global Finance Journal*, 3:79–95, 1992.

[39] M. A. Croon. The axiomatization of additive difference models for preference judgements. In E. Degreef and G. van Buggenhaut, editors, *Trends in Mathematical Psychology*, pages 193–227, Amsterdam, 1984. North-Holland.

[40] V. Dahlstrand and H. Montgomery. Information search and evaluation processes in decision-making: A computer-based process tracking study. *Acta Psychologica*, 56:113–123, 1984.

[41] P. de Donder, M. Le Breton, and M. Truchon. Choosing from a weighted tournament. *Mathematical Social Sciences*, 40:85–109, 2000.

[42] G. Debreu. Representation of a preference ordering by a numerical function. In R. Thrall, C. H. Coombs, and R. Davies, editors, *Decision Processes*, pages 159–175, New York, 1954. Wiley.

[43] G. Debreu. *Theory of value: An axiomatic analysis of economic equilibrium*. John Wiley & Sons, New York, 1959.

[44] G. Debreu. Topological methods in cardinal utility theory. In K. J. Arrow, S. Karlin, and P. Suppes, editors, *Mathematical methods in the Social Sciences*, pages 16–26. Stanford University Press, Stanford, 1960.

[45] G. Debreu. Continuity properties of Paretian utility. *International Economic Review*, 5(3):285–293, September 1964.

[46] Ph. Delquié. Inconsistent trade-offs between attributes: New evidence in preference assessment biases. *Management Science*, 39(11):1382–1395, November 1993.

[47] A. I. Dimitras, C. Zopounidis, and C. Hurson. Assessing financial risks using a multicriteria sorting procedure: The case of country risk assessment. *Foundations of Computing and Decision Sciences*, 29:97–109, 2001.

[48] M. Doumpos, S. H. Zanakis, and C. Zopounidis. Multicriteria preference disaggregation for classification problems with an application to global investing risk. *Decision Sciences*, 32(2):333–385, 2001.

[49] D. Dubois, J.-L. Marichal, H. Prade, M. Roubens, and R. Sabbadin. The use of the discrete Sugeno integral in decision-making: A survey. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(5):539–561, 2001.

[50] J. S. Dyer and R. K. Sarin. Measurable multiattribute value functions. *Operations Research*, 27:810–822, 1979.

[51] W. Edwards. Social utilities. *Engineering Economist*, 6:119–129, 1971.

[52] W. Edwards. How to use multiattribute utility measurement for social decision making. *IEEE Transactions on Systems, Man and Cybernetics*, 7(5):326–340, 1977.

[53] W. Edwards and F. Hutton Barron. SMART and SMARTER: Improved simple methods for multiattribute utility measurement. *Organizational Behavior and Human Decision Processes*, 60:306–325, 1994.

[54] J.-C. Falmagne. On a class of probabilistic conjoint measurement models: Some diagnostic properties. *Journal of Mathematical Psychology*, 19(2):73–88, 1979.

[55] H. Fargier and P. Perny. Modélisation des préférences par une règle de concordance généralisée. In A. Colorni, M. Paruccini, and B. Roy, editors, A-MCD-A, *Aide Multicritère à la Décision/Multiple Criteria Decision Aid*, pages 99–115. European Commission, Joint Research Centre, 2001.

[56] Final report of the British Association for the Advancement of Science. 2:331–349, 1940.

[57] P. C. Fishburn. Intransitive indifference in preference theory: A survey. *Operations Research*, 18(2):207–228, 1970.

[58] P. C. Fishburn. *Utility theory for decision-making*. Wiley, New York, 1970.

[59] P. C. Fishburn. Lexicographic orders, utilities and decision rules: A survey. *Management Science*, 20(11):1442–1471, 1974.

[60] P. C. Fishburn. Axioms for lexicographic preferences. *Review of Economic Studies*, 42:415–419, 1975.

[61] P. C. Fishburn. Noncompensatory preferences. *Synthese*, 33:393–403, 1976.

[62] P. C. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33:469–489, 1977.

[63] P. C. Fishburn. A survey of multiattribute/multicriteria evaluation theories. In S. Zionts, editor, *Multicriteria problem solving*, pages 181–224. Springer Verlag, Berlin, 1978.

[64] P. C. Fishburn. Lexicographic additive differences. *Journal of Mathematical Psychology*, 21:191–218, 1980.

[65] P. C. Fishburn. Nontransitive measurable utility. *Journal of Mathematical Psychology*, 26:31–67, 1982.

[66] P. C. Fishburn. *Interval orders and intervals graphs*. Wiley, New York, 1985.

[67] P. C. Fishburn. *Nonlinear preference and utility theory*. Johns Hopkins University Press, Baltimore, 1988.

[68] P. C. Fishburn. Normative theories of decision making under risk and under uncertainty. In *Nonconventional preference relations in decision making*, pages 1–21. Springer, Berlin, 1988.

[69] P. C. Fishburn. Additive non-transitive preferences. *Economic Letters*, 34:317–321, 1990.

[70] P. C. Fishburn. Continuous nontransitive additive conjoint measurement. *Mathematical Social Sciences*, 20:165–193, 1990.

[71] P. C. Fishburn. Skew symmetric additive utility with finite states. *Mathematical Social Sciences*, 19:103–115, 1990.

[72] P. C. Fishburn. Nontransitive additive conjoint measurement. *Journal of Mathematical Psychology*, 35:1–40, 1991.

[73] P. C. Fishburn. Nontransitive preferences in decision theory. *Journal of Risk and Uncertainty*, 4:113–134, 1991.

[74] P. C. Fishburn. Additive differences and simple preference comparisons. *Journal of Mathematical Psychology*, 36:21–31, 1992.

[75] P. C. Fishburn. A general axiomatization of additive measurement with applications. *Naval Research Logistics*, 39(6):741–755, 1992.

[76] P. C. Fishburn. On nonstandard nontransitive additive utility. *Journal of Economic Theory*, 56:426–433, 1992.

[77] P. C. Fishburn. Finite linear qualitative probability. *Journal of Mathematical Psychology*, 40:21–31, 1996.

[78] P. C. Fishburn. Cancellation conditions for multiattribute preferences on finite sets. In M. H. Karwan, J. Spronk, and J. Wallenius, editors, *Essays in Decision Making*, pages 157–167, Berlin, 1997. Springer Verlag.

[79] S. French. *Decision theory – An introduction to the mathematics of rationality*. Ellis Horwood, London, 1993.

[80] D. G. Fryback and R. L. Keeney. Constructing a complex judgmental model: An index of trauma severity. *Management Science*, 29:869–883., 1983.

[81] G. Furkhen and M. K. Richter. Additive utility. *Economic Theory*, 1:83–105, 1991.

[82] G. Furkhen and M. K. Richter. Polynomial utility. *Economic Theory*, 1:231–249, 1991.

[83] D. Gale. *The theory of linear economic models*. McGraw-Hill, New York, 1960.

[84] I. Gilboa and R. Lapson. Aggregation of semiorders: Intransitive indifference makes a difference. *Economic Theory*, 5:109–126, 1995.

[85] Ch. Gonzales. Additive utilities when some components are solvable and others not. *Journal of Mathematical Psychology*, 40:141–151, 1996.

[86] Ch. Gonzales. *Utilités additives : Existence et construction*. Thèse de doctorat, Université Paris 6, Paris, France, 1996.

[87] Ch. Gonzales. Two factor additive conjoint measurement with one solvable component. *Journal of Mathematical Psychology*, 44:285–309, 2000.

[88] Ch. Gonzales. Additive utility without restricted solvavibility on every component. *Journal of Mathematical Psychology*, 47:47–65, 2003.

[89] W. M. Gorman. The structure of utility functions. *Review of Economic Studies*, XXXV:367–390, 1968.

[90] M. Grabisch. The application of fuzzy integrals to multicriteria decision making. *European Journal of Operational Research*, 89:445–456, 1996.

[91] M. Grabisch. Fuzzy integral as a flexible and interpretable tool of aggregation. In *Aggregation and fusion of imperfect information*, pages 51–72. Physica, Heidelberg, 1998.

[92] M. Grabisch, Ch. Labreuche, and J.-C. Vansnick. On the extension of pseudo-boolean functions for the aggregation of interacting criteria. *European Journal of Operational Research*, 148(1):28–47, 2003.

[93] M. Grabisch, S. A. Orlovski, and R. R. Yager. Fuzzy aggregation of numerical preferences. In R. Słowiński, editor, *Fuzzy sets in decision analysis, operations research and statistics*, pages 31–68. Kluwer, Boston, MA, 1998.

[94] M. Grabisch and P. Perny. Agrégation multicritère. In B. Bouchon-Meunier and C. Marsala, editors, *Logique floue, Principes, Aide à la décision*, IC^2, pages 82–120. Hermès, 2002.

[95] M. Grabisch and M. Roubens. Application of the Choquet integral in multicriteria decision making. In M. Grabisch, T. Murofushi, and M. Sugeno, editors, *Fuzzy measures and integrals*, pages 348–374. Physica Verlag, Heidelberg, 2000.

[96] N. Grassin. Constructing criteria population for the comparison of different options of high voltage line routes. *European Journal of Operational Research*, 26:42–47, 1886.

[97] S. Greco, B. Matarazzo, and R. Słowiński. Rough approximation of a preference relation by dominance relations. *European Journal of Operational Research*, 117:63–83, 1999.

[98] S. Greco, B. Matarazzo, and R. Słowiński. The use of rough sets and fuzzy sets in MCDM. In T. Gal, T. Hanne, and T. Stewart, editors, *Multicriteria decision making, Advances in* MCDM *models, algorithms, theory and applications*, pages 14.1–14.59, Dordrecht, 1999. Kluwer.

[99] S. Greco, B. Matarazzo, and R. Słowiński. Axiomatic basis of noncompensatoty preferences. Communication to *FUR X* (Foundations of Utility and Risk Theory), 30 May–2 June, Torino, Italy, 2001.

[100] S. Greco, B. Matarazzo, and R. Słowiński. Conjoint measurement and rough set approach for multicriteria sorting problems in presence of ordinal criteria. In A. Colorni, M. Paruccini, and B. Roy, editors, A-MCD-A, *Aide Multicritère à la Décision/Multiple Criteria Decision Aid*, pages 117–144. European Commission, Joint Research Centre, 2001.

[101] S. Greco, B. Matarazzo, and R. Słowiński. Rough sets theory for multicriteria decision analysis. *European Journal of Operational Research*, 129:1–47, 2001.

[102] S. Greco, B. Matarazzo, and R. Słowiński. Preference representation by means of conjoint measurement and decision rule model. In D. Bouyssou, É. Jacquet-Lagrèze, P. Perny, R. Sł owiński, D. Vanderpooten, and Ph. Vincke, editors, *Aiding Decisions with Multiple Criteria: Essays in Honour of Bernard Roy*, pages 263–313. Kluwer, Dordrecht, 2002.

[103] P. E. Green and V. Srinivasan. Conjoint analysis in consumer research: Issues and outlook. *Journal of Consumer Research*, 5:103–152, 1978.

[104] P. E. Green, D. S. Tull, and G. Albaum. *Research for marketing decisions*. Englewood Cliffs, 1988.

[105] E. Grigoroudis, Y. Siskos, and O. Saurais. TELOS: A customer satisfaction evaluation software. *Computers and Operations Research*, 27(7–8):799–817, 2000.

[106] J. Guild. Part III of Quantitative estimation of sensory events. Interim report, British Academy for the Advancement of Science, 1936. pp. 296–328.

[107] F. Gul. Savage's theorem with a finite number of states. *Journal of Economic Theory*, 57:99–110, 1992.

[108] G. Iverson and J.-C. Falmagne. Statistical issues in measurement. *Mathematical Social Sciences*, 10:131–153, 1985.

[109] É. Jacquet-Lagrèze. Interactive assessment of preferences using holistic judgments. The PREFCALC system. In C. A. Bana e Costa, editor, *Readings in multiple criteria decision aid*, pages 335–350. Springer Verlag, Berlin, 1990.

[110] É. Jacquet-Lagrèze. An application of the UTA discriminant model for the evaluation of R&D projects. In P. Pardalos, Y. Siskos, and C. Zopounidis, editors, *Advances in Multicriteria Analysis*, pages 203–211. Kluwer Academic Publishers, Dordrecht, 1995.

[111] É. Jacquet-Lagrèze and J. Siskos. Assessing a set of additive utility functions for multicriteria decision making: The UTA method. *European Journal of Operational Research*, 10:151–164, 1982.

[112] J.-Y. Jaffray. *Existence, propriétés de continuité, additivité de fonctions d'utilité sur un espace partiellement ou totalement ordonné*. Thèse de doctorat d'état, Université Paris 6, Paris, France, 1974.

[113] J.-Y. Jaffray. On the extension of additive utilities to infinite sets. *Journal of Mathematical Psychology*, 11:431–452, 1974.

[114] R. M. Johnson. Trade-off analysis of consumer values. *Journal of Marketing Research*, 11:121–127, 1974.

[115] E. Karni and Z. Safra. The hexagon condition and additive representation for two dimensions: An algebraic approach. *Journal of Mathematical Psychology*, 42:393–399, 1998.

[116] R. L. Keeney. Measurement scales for quantifying attributes. *Behavioral Science*, 26:29–36, 1981.

[117] R. L. Keeney. Building models of values. *European Journal of Operational Resesearch*, 37(2):149–157, 1988.

[118] R. L. Keeney. Structuring objectives for problems of public interest. *Operations Research*, 36:396–405, 1988.

[119] R. L. Keeney. *Value-focused thinking*. Harvard University Press, Cambridge, MA., 1992.

[120] R. L. Keeney, J. S. Hammond, and H. Raiffa. *Smart choices: A guide to making better decisions*. Harvard University Press, Boston, 1999.

[121] R. L. Keeney and H. Raiffa. *Decisions with multiple objectives: Preferences and value tradeoffs*. Wiley, New York, 1976.

[122] R. L. Keeney and G. A. Robillard. Assessing and evaluating environ-mental impacts at proposed nuclear power plant sites. *Journal of Environmental Economics and Management*, 4:153–166, 1977.

[123] V. Köbberling. Strenth of preference and cardinal utility. Working paper, University of Maastricht, Maastricht, June 2002.

[124] V. Köbberling. Comment on: Edi Karni & Zvi Safra (1998) The hexagon condition and additive representation for two dimensions: An algebraic approach. *Journal of Mathematical Psychology*, 47(3):370, 2003.

[125] T. C. Koopmans. Stationary ordinal utility and impatience. *Econometrica*, 28:287–309, 1960.

[126] T. C. Koopmans. Representation of prefernce orderings over time. In C. B. McGuire and R. Radner, editors, *Decision and Organization*, pages 57–100. Noth-Holland, Amsterdam, 1972.

[127] D. H. Krantz. Conjoint measurement: The Luce-Tukey axiomatization and some extensions. *Journal of Mathematical Psychology*, 1:248–277, 1964.

[128] D. H. Krantz. Extensive measurement in semiorders. *Philosophy of Science*, 34:348–362, 1967.

[129] D. H. Krantz, R. D. Luce, P. Suppes, and A. Tversky. *Foundations of measurement*, vol. 1: *Additive and polynomial representations*. Academic Press, New York, 1971.

[130] G. Laffond, J.-F. Laslier, and M. Le Breton. Condorcet choice correspondences: A set-theoretical comparison. *Mathematical Social Sciences*, 30:23–36, 1995.

[131] J.-F. Laslier. *Tournament solutions and majority voting*. Springer Verlag, Berlin, 1997.

[132] J. Louvière. *Analyzing Decision Making: Metric Conjoint Analysis*. Sage, Park, CA, 1988.

[133] R. D. Luce. Semiorders and a theory of utility discrimination. *Econometrica*, 24:178–191, 1956.

[134] R. D. Luce. Three axiom systems for additive semiordered structures. *SIAM Journal of Applied Mathematics*, 25:41–53, 1973.

[135] R. D. Luce. Conjoint measurement: A brief survey. In David E. Bell, Ralph L. Keeney, and Howard Raiffa, editors, *Conflicting objectives in decisions*, pages 148–171. Wiley, New York, 1977.

[136] R. D. Luce. Lexicographic tradeoff structures. *Theory and Decision*, 9:187–193, 1978.

[137] R. D. Luce. *Utility of gains and losses: Measurement-theoretical and experimental approaches*. Lawrence Erlbaum Publishers, Mahwah, New Jersey, 2000.

[138] R. D. Luce and M. Cohen. Factorizable automorphisms in solvable conjoint structures. I. *Journal of Pure and Applied Algebra*, 27(3):225–261, 1983.

[139] R. D. Luce, D. H. Krantz, P. Suppes, and A. Tversky. *Foundations of measurement*, vol. 3: *Representation, axiomatisation and invariance*. Academic Press, New York, 1990.

[140] R. D. Luce and A. A. J. Marley. Extensive measurement when concatenation is restricted. In S. Morgenbessser, P. Suppes, and M. G. White, editors, *Philosophy, science and method: Essays in honor of Ernest Nagel*, pages 235–249. St. Martin’s Press, New York, 1969.

[141] R. D. Luce and J. W. Tukey. Simultaneous conjoint measurement: A new type of fundamental measurement. *Journal of Mathematical Psychology*, 1:1–27, 1964.

[142] A. Maas and P. P. Wakker. Additive conjoint measurement for multiattribute utility. *Journal of Mathematical Psychology*, 38:86–101, 1994.

[143] J.-L. Marichal. An axiomatic approach of the discrete Choquet integral as a tool to aggregate interacting criteria. *IEEE Transactions on Fuzzy Systems*, 8, 2000.

[144] J.-L. Marichal. On Choquet and Sugeno integrals as aggregation functions. In M. Grabisch, T. Murofushi, and M. Sugeno, editors, *Fuzzy measures and integrals*, pages 247–272. Physica Verlag, Heidelberg, 2000.

[145] J.-L. Marichal. On Sugeno integralsas an aggregation function. *Fuzzy Sets and Systems*, 114, 2000.

[146] J.-L. Marichal and M. Roubens. Determination of weights of interacting criteria from a reference set. *European Journal of Operational Research*, 124:641–50, 2000.

[147] K. O. May. Intransitivity, utility and the aggregation of preference patterns. *Econometrica*, 22:1–13, 1954.

[148] G. Mihelis, E. Grigoroudis, Y. Siskos, Y. Politis, and Y. Malandrakis. Customer satisfaction measurement in the private bank sector. *European Journal of Operational Research*, 130(2):347–360, April 2001.

[149] N. R. Miller. Graph theoretical approaches to the theory of voting. *American Journal of Political Science*, 21:769–803, 1977.

[150] J. Miyamoto and P. P. Wakker. Multiattribure utility theory without expected utility foundations. *Operations Research*, 44(2):313–326, 1996.

[151] H. Montgomery. A study of intransitive preferences using a think aloud procedure. In H. Jungerman and G. de Zeeuw, editors, *Decision-making and Change in Human Affairs*, pages 347–362, Dordrecht, 1977. D. Reidel.

[152] H. Montgomery and O. Svenson. On decision rules and information processing strategies for choice among multiattribute alternatives. *Scandinavian Journal of Psychology*, 17:283–291, 1976.

[153] Y. Nakamura. Lexicographic additivity for multi-attribute preferences on finite sets. *Theory and Decision*, 42:1–19, 1997.

[154] Y. Nakamura. Additive utility on densely ordered sets. *Journal of Mathematical Psychology*, 46:515–530, 2002.

[155] L. Narens. *Abstract measurement theory*. MIT press, Cambridge, Mass., 1985.

[156] L. Narens and R. D. Luce. The algebra of measurement. *Journal of Pure and Applied Algebra*, 8(2):197–233, 1976.

[157] A. Ostanello. Action evaluation and action structuring – Different decision aid situations reviewed through two actual cases. In C. A. Bana e Costa, editor, *Readings in multiple criteria decision aid*, pages 36–57. Springer Verlag, Berlin, 1990.

[158] J. E. Peris and B. Subiza. Condorcet choice correspondences for weak tournaments. *Social Choice and Welfare*, 16:217–231, 1999.

[159] J. Pfanzagl. *Theory of measurement*. Physica Verlag, Würzburg, 2nd revised edition edition, 1971.

[160] M. Pirlot and Ph. Vincke. Lexicographic aggregation of semiorders. *Journal of Multicriteria Decision Analysis*, 1:47–58, 1992.

[161] M. Pirlot and Ph. Vincke. *Semiorders. Properties, representations, applications*. Kluwer, Dordrecht, 1997.

[162] J.-Ch. Pomerol and S. Barba-Romero. *Multicriterion Decision in Management, Principles and Practice*. Kluwer, Dordrecht, 2000.

[163] M. Pöyhönen, H. Vrolijk, and R. P. Hämäläinen. Behavioral and procedural consequences of structural variations in value trees. *European Journal of Operational Research*, 134(1):216–227, October 2001.

[164] H. Raiffa. *Decision analysis – Introductory lectures on choices under uncertainty*. Addison-Wesley, Reading, Mass., 1968.

[165] H. Raiffa. Preference for multi-attributed alternatives. RAND Memorandum, RM-5868-DOT/RC, December 1968.

[166] M. K. Richter. Rational choice and polynomial measurement theory. *Journal of Mathematical Psychology*, 12:99–113, 1975.

[167] F. S. Roberts. *Measurement theory with applications to decision making, utility and the social sciences*. Addison-Wesley, Reading, 1979.

[168] F. S. Roberts and R. D. Luce. Axiomatic thermodynamics and extensive measurement. *Synthese*, 18:311–326, 1968.

[169] M. Roubens and Ph. Vincke. *Preference modelling*. Springer Verlag, Berlin, 1985.

[170] B. Roy. Classement et choix en présence de points de vue multiples (la méthode ELECTRE). *RIRO*, 2:57–75, 1968.

[171] B. Roy. Main sources of inaccurate determination, uncertainty and imprecision in decision models. *Mathematical and Computer Modelling*, 12(10–11):1245–1254, 1989.

[172] B. Roy. The outranking approach and the foundations of ELECTRE methods. *Theory and Decision*, 31:49–73, 1991.

[173] B. Roy. *Multicriteria methodology for decision aiding*. Kluwer, Dordrecht, 1996. Original version in French "*Méthodologie multicritère d'aide à la décision*", Economica, Paris, 1985.

[174] B. Roy and D. Bouyssou. *Aide multicritère à la décision : méthodes et cas*. Economica, Paris, 1993.

[175] J. E. Russo and B. A. Dosher. Strategies for multiattribute binary choice. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 9:676–696, 1983.

[176] D. G. Saari. Connecting and resolving Sen's and Arrow's theorems. *Social Choice and Welfare*, 15:239–261, 1998.

[177] L. J. Savage. *The Foundations of Statistics*. John Wiley and Sons, New York, 1954.

[178] T. Schwartz. *The logic of collectice choice*. Columbia University Press, 1986.

[179] D. Scott. Measurement structures and linear inequalities. *Journal of Mathematical Psychology*, 1:233–247, 1964.

[180] D. Scott and P. Suppes. Foundational aspects of theories of measurement. *Journal of Symbolic Logic*, 23:113–128, 1958.

[181] U. Segal. A sufficient condition for additively separable functions. *Journal of Mathematical Economics*, 23:295–303, 1994.

[182] U. Segal and J. Sobel. Min, Max and Sum. *Journal of Economic Theory*, 106:126–150, 2002.

[183] A. K. Sen. Social choice theory. In K. J. Arrow and M. D. Intriligator, editors, *Handbook of mathematical economics*, volume 3, pages 1073–1181. North-Holland, Amsterdam, 1986.

[184] L. Shapiro. Conditions for expected uility maximization. *Annals of Statistics*, 7:1288–1302, 1979.

[185] Y. Siskos. Evaluating a system of furniture retail outlets using an interactive ordinal regression model. *European Journal of Operational Research*, 23:179–193, 1986.

[186] Y. Siskos and D. K. Despotis. A DSS oriented method for multiobjective linear programming problems. *Decision Support Systems*, 5(1):47–56, 1989.

[187] Y. Siskos, D. K. Despotis, and M. Ghediri. Multiobjective modelling for regional agricultural planning: Case study in Tunisia. *European Journal of Operational Research*, 77(3):375–391, 1993.

[188] Y. Siskos, E. Grigoroudis, C. Zopounidis, and O. Saurais. Measuring customer satisfaction using a collective preference disaggregation model. *Journal of Global Optimization*, 12(2):175–195, 1998.

[189] Y. Siskos, N. F. Matsatsinis, and G. Baourakis. Multicriteria analysis in agricultural marketing: The case of French olive oil market. *European Journal of Operational Research*, 130(2):315–331, April 2001.

[190] Y. Siskos, A. Spyridakos, and D. Yannacopoulos. Using artificial intelligence and visual techniques into preference disaggregation analysis: The MIIDAS system. *European Journal of Operational Research*, 113(2):281–299, March 1999.

[191] Y. Siskos and D. Yannacopoulos. UTASTAR, An ordinal regression method for building additive value functions. *Investigaçao Operacional*, 5(1):39–53, 1985.

[192] Y. Siskos and C. Zopounidis. The evaluation of venture capital investment activity: An interactive assessment. *European Journal of Operational Research*, 31(3):304–313, 1987.

[193] H. J. Skala. *Non-Archimedean Utility Theory*. Kluwer, Dordrecht, 1975.

[194] J. Sounderpandian. Value functions when decision criteria are not totally substitutable. *Operations Research*, 39:592–600, 1991.

[195] A. Spyridakos, Y. Siskos, D. Yannacopoulos, and A. Skouris. Multicriteria job evaluation for large organizations. *European Journal of Operational Research*, 130(2):375–387, April 2001.

[196] T. Stewart. An interactive multiple objective linear programming method based on piecewise linear additive value functions. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-17(5):799–805, 1987.

[197] W. G. Stillwell, D. von Winterfeldt, and R. S. John. Comparing hierarchical and nonhierarchical weighting methods for eliciting multiattribute value models. *Management Science*, 33:442–50, 1987.

[198] P. Suppes, D. H. Krantz, R. D. Luce, and A. Tversky. *Foundations of measurement*, vol. 2: *Geometrical, threshold, and probabilistic representations*. Academic Press, New York, 1989.

[199] P. Suppes and M. Winet. An axiomatization of utility based on the notion of utility difference. *Management Science*, 1:259–270, 1955.

[200] O. Svenson. Process description of decision making. *Organizational Behavior and Human Performance*, 23:86–112, 1979.

[201] R. J. Titiev. Measurement structures in classes that are not universally axiomatizable. *Journal of Mathematical Psychology*, 9:200–205, 1972.

[202] A. Tversky. A general theory of polynomial conjoint measurement. *Journal of Mathematical Psychology*, 4:1–20, 1967.

[203] A. Tversky. Intransitivity of preferences. *Psychological Review*, 76:31–48, 1969.

[204] Ph. Vincke. *Multi-criteria decision aid*. Wiley, New York, 1992. Original version in French "*L'Aide Multicritère à la Décision*", Éditions de l'Université de Bruxelles-Éditions Ellipses, Brussels, 1989.

[205] Ph. Vincke. Outranking approach. In T. Gal, T. Stewart, and T. Hanne, editors, *Multicriteria decision making, Advances in* MCDM *models, algorithms, theory and applications*, pages 11.1–11.29. Kluwer, 1999.

[206] K. Vind. Independent preferences. *Journal of Mathematical Economics*, 20:119–135, 1991.

[207] J. von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Wiley, 2nd edition, 1947.

[208] R. von Nitzsch and M. Weber. The effect of attribute ranges on weights in multiattribute utility measurements. *Management Science*, 39(8):937–943, August 1993.

[209] D. von Winterfeldt and W. Edwards. *Decision analysis and behavioral research*. Cambridge University Press, Cambridge, 1986.

[210] P. P. Wakker. Cardinal coordinate independence for expected utility. *Journal of Mathematical Psychology*, 28(1):110–117, 1984.

[211] P. P. Wakker. *Additive representations of preferences: A new foundation of decision analysis*. Kluwer, Dordrecht, 1989.

[212] P. P. Wakker. Additive representation for equally spaced structures. *Journal of Mathematical Psychology*, 35:260–266, 1991.

[213] P. P. Wakker. Additive representations of preferences, A new foundation of decision analysis; The algebraic approach. In J.-P. Doignon and J.-C. Falmagne, editors, *Mathematical Psychology: Current Developments*, pages 71–87, Berlin, 1991. Springer Verlag.

[214] P. P. Wakker. Additive representations on rank-ordered sets. I. The algebraic approach. *Journal of Mathematical Psychology*, 35(4):501–531, 1991.

[215] P. P. Wakker. Additive representations on rank-ordered sets. II. The topological approach. *Journal of Mathematical Economics*, 22(1):1–26, 1993.

[216] M. Weber, F. Eisenfuhr, and D. von Winterfeld. The effects of splitting attributes on weights in multiattribute utility measurement. *Management Science*, 34(4):431–445, 1988.

[217] J. A. Weymark. Generalized Gini inequality indices. *Mathematical Social Sciences*, 1:409–430, 1981.

[218] U. Wille. Linear measurement models: Axiomatizations and axiomatizability. *Journal of Mathematical Psychology*, 44:617–650, 2000.

[219] C. Zopounidis and M. Doumpos. Stock evaluation using a preference disaggregation methodology. *Decision Science*, 30:313–336, 1999.

[220] C. Zopounidis and M. Doumpos. Building additive utilities for multi-group hierarchical discrimination: The MHDIS method. *Optimization Methods & Software*, 14(3):219–240, 2000.

[221] C. Zopounidis and M. Doumpos. PREFDIS: A multicriteria decision support system for sorting decision problems. *Computers & Operations Research*, 27(7–8):779–797, June 2000.

[222] C. Zopounidis and M. Doumpos. Multicriteria preference disaggregation for classification problems with an application to global investing risk. *Decision Sciences*, 32(2):333–385, 2001.

[223] C. Zopounidis and M. Doumpos. A preference disaggregation decision support system for financial classification problems. *European Journal of Operational Research*, 130(2):402–413, April 2001.

Towards a Spatial Decision Support System: Multi-Criteria Evaluation Functions Inside Geographical Information Systems

Salem Chakhar[*] and Jean-Marc Martel[†]

Résumé

L'objectif de cet article est de présenter une stratégie d'intégration des systèmes d'information géographiques (SIG) et de l'analyse multicritère (AMC), une famille d'outils de la recherche opérationnelle/science de management (RO/SM) qui connaît des applications réussies dans différents domaines depuis les années 1960. En effet, le SIG présente plusieurs limites dans le domaine d'aide à la décision spatiale. La solution à ces limites consiste à intégrer le SIG avec différents outils de la RO/SM et spécialement avec l'AMC. Le but à long terme de cette intégration vise le développement d'un système informatique d'aide à la décision (SIAD) spatiale. A cet effet, un design d'un SIAD spatiale est également proposé dans ce papier.

Mots-clefs : Système d'information géographique, Analyse multicritère, Aide à la decision spatiale, Système informatique d'aide à la décision spatiale.

Abstract

The essence of this paper is to present a strategy for integrating geographical information systems (GIS) and multicriteria analysis (MCA), a family of operational research/management science (OR/MS) tools that have experienced very successful applications in different domains since the 1960s. In fact, GIS has several limitations in the domain of spatial decision-aid. The remedy to these limitations is to integrate GIS technology with OR/MS tools and especially with MCA. The long-term aim of such an integration is to develop the so-called spatial decision support system (SDSS), which is devoted to help deciders in spatially-related problems. Thus, a design of a SDSS is also presented in this paper.

Key words : Geographical information system, Multicriteria analysis, Spatial decision-aid, Spatial decision support system.

[*] LAMSADE, Université Paris Dauphine, Place du Maréchal de Lattre de Tassigny F-75775 Paris Cedex 16, France. *Email:* `chakhar@lamsade.dauphine.fr`

[†] Faculté des Sciences de l'Administration, Pavillon Palasis-Prince, Université Laval, Québec G1K 7P4, Canada. *Email:* `Jean-Marc.Martel@fsa.ulaval.ca`

1 Introduction

A fully functional GIS is a smooth integration of several components and different subsystems (for more information concerning GIS technology, see, for e.g., Burrough and McDonnell (1998)). It is devoted especially to collect, store, retrieve and analyze spatially-referenced data. Even though numerous practical applications have shown that GIS is a powerful tool of acquisition, management and analysis of spatially-referenced data, the impression shared by most of current OR/MS specialists (e.g. Janssen and Rietveld, 1990; Carver, 1991; Fischer and Nijkamp, 1993; Laaribi et *al*., 1993, 1996; Malczewski, 1999; Laaribi, 2000) and decision-makers sets the GIS to be a limited tool in spatial decision-aid domain. This due essentially to its lack in more powerful analytic tools enabling it to deal with spatial problems, where it is usually several parties having conflicting objectives are involved in the decision-making process.
Among the critics that have been addressed to GIS technology, we enumerate the following ones (Burrough, 1990; Janssen and Rietveld, 1990; Carver, 1991; Goodchild, 1992; Laaribi et *al*., 1993; Laaribi, 2000):

- decision maker's preferences are not taken into account by current GISs. Some raster-based GISs, however, allow ratios for criteria (e.g. the models GRID Arc/Info of ESRI and MGE Grid Analyst d'Intergraph) but these ratios are usually introduced prior to the solution(s) generation process, i.e., in a non-interactive manner,

- in most GIS packages spatial analytical functionalities lie mainly in the ability to perform deterministic overlay and buffer operations which are of limited use when multiple and conflicting criteria are concerned,

- current GIS do not permit the assessment and comparison of different scenarios. They identify only solutions satisfying all criteria simultaneously,

- analytic functionalities found in most GIS are oriented towards the management of data but not towards an effective analysis of them,

- overlapping technique that is found in nearly all standard GIS becomes difficult to comprehend when the number of layers is more than four or five. Moreover, overlapping methods consider that all features are of equal importance.

The remedy that has been supported by different researchers consists in integrating the GIS with different OR/MS tools. Practically, the idea of integrating GIS with several OR/MS tools seems to be a long-term solution. In fact, this requires the development of a coherent theory of spatial analysis parallel to a theory of spatial data (Laaribi, 2000). A more realistic solution is, however, to incorporate only a family of analyze methods into the GIS. Intuitively, the most suitable family is that of MCA, which is a family of OR/MS tools that have experienced very successful applications in different domains since the 1960s (section §2 provides a brief description of MCA. More information on the subject are available in, for e.g., Roy 1985; Vincke, 1992; Pomerol and Barba-Romero, 1993; Roy and Bouyssou, 1993; Belton and Stewart, 2002).

Perhaps, the most convincing argument that supports the idea of GIS-MCA integration is related to the complementarity of the two tools. In fact, the former is a powerful tool for managing spatially-referenced data, while the latter is an efficient tool for modelling spatial problems. Another important argument consists of the ability of MCA to support efficiently the different phases of the Simon's (1960) decision-making process phases.

The remaining of this paper is structured as follows. Section 2 presents MCA paradigm. Then, section 3 provides two solutions for extending the capabilities of GIS. The first long-term perspective solution looks to the development of the SDSS. The second short-term perspective solution focalizes on GIS-MCA integration. The proposed short-term strategy for integrating GIS and MCA is then detailed in section 4. Lastly, some concluding remarks are given in section 5.

2 Multicriteria analysis paradigm

It is quite difficult to define precisely MCA. However, various definitions appear in literature. One common definition is that of Roy (1985). Roy postulates that MCA is '*a decision-aid and a mathematical tool allowing the comparison of different alternatives or scenarios according to many criteria, often contradictory, in order to guide the decider(s) towards a judicious choice*'.

Whatever the definition, it is generally assumed in MCA that the decider-maker (DM) has to choose among several possibilities, called *actions* or *alternatives*[1]. The *set of alternatives* is the collection of all alternatives. Selecting an alternative among this set depends on many characteristics, often contradictory, called *criteria*. Accordingly, the decision-maker will generally have to be content with a *compromising solution*.

The multicriteria problems are commonly categorized as *continuous* or *discrete*, depending on the domain of alternatives (Zanakis et *al*., 1998). Hwang and Yoon (1981) classify them as (i) multiple attribute decision-making (MADM), and (ii) multiple objective decision-making (MODM). According to Zanakis et *al.* (1998), the former deals with discrete, usually limited, number of pre-specified alternatives. The latter deals with variable decision values to be determined in a continuous or integer domain of infinite or large number of choices.

[1] The two terms 'action' and 'alternative' are slightly different. In fact, the term 'alternative' applies when actions are mutually exclusive, i.e., selecting an action excludes any other one. In practice, however, we may be called to choose a combination of several actions, which violates the exclusion hypothesis. In this paper, we adopt the term 'alternative'.

Different MCA models have been developed during the second half of the 20*th* century (see Vincke (1992), Pomerol and Barba-Romero (1993), Roy and Bouyssou (1993), Maystre et *al*. (1994) and Belton and Stewart (2002) for some MCA models). They essentially differ from each other in the nature of the aggregation procedure, i.e., the manner in which different alternatives are globally evaluated. However, they may be categorized into two general models of MADM (a) and MODM (b) (Figure 1). These models, which will be detailed in the two next sections, illustrate how the different elements of the decision problem are linked to each other. They are inspired from the MADM general model of Jankowski (1995).

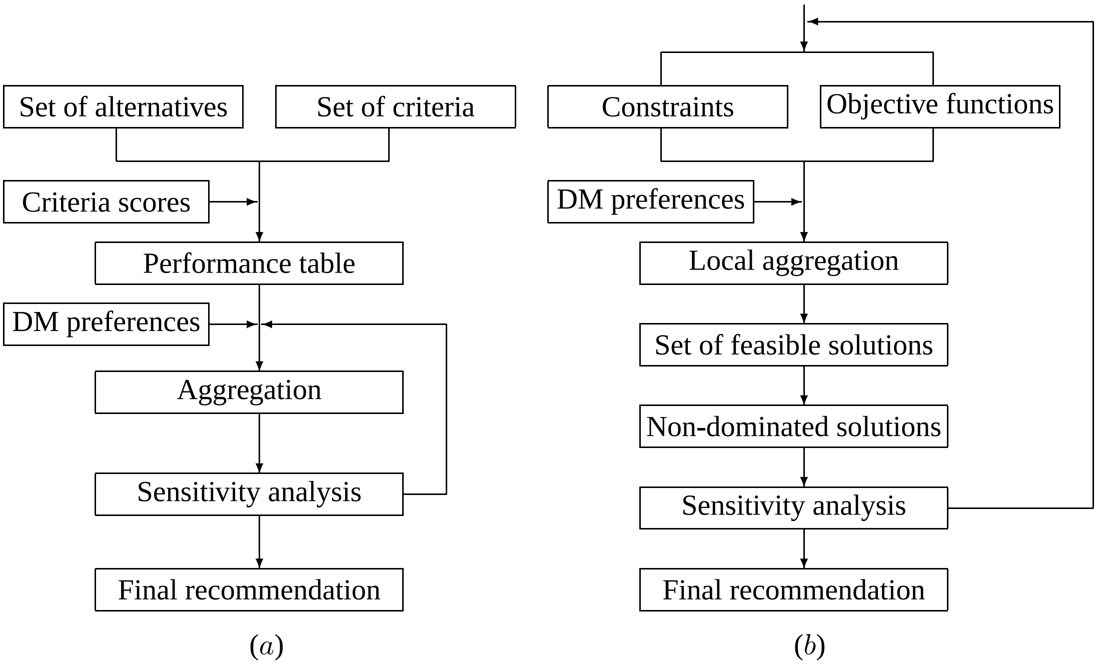


Figure 1: The MADM (a) and MODM (b) general models

2.1 Multiple attribute decision-making general model

The first requirement of nearly all MADM techniques is a *performance table* containing the *evaluations* or *criteria scores* of a *set of alternatives* on the basis of a *set of criteria*. Generally, the two sets are separately defined. An efficient conception of the decision problem necessitates, however, that criteria and alternatives are jointly defined. The next step in MADM consists in the *aggregation* of the different criteria scores using a specific *aggregation procedure* and taking into account the DM *preferences*, generally represented in terms of *weights* that are assigned to different criteria. The aggregation of criteria scores permits the DM to make comparison between the different alternatives on the basis of these scores. Aggregation procedures are somehow the identities of the MCA

techniques. In MADM, they are usually categorized into two great families (i) outranking relation-based family, and (ii) utility function-based family (see Vincke (1992)).
In addition to the weights, the DM's preferences may also take the form of *aspiration levels* or *cut-of values*. The aspiration level represents the degree of performance according to a given criterion making the DM fully satisfied with an alternative in regard to the considered criterion, while the cut-of value represents the degree of performance which ought to be attained (or exceeded) by an alternative; otherwise, it is rejected.
The uncertainty and the fuzziness generally associated with any decision situation require a *sensitivity analysis* enabling the decider(s) to test the consistency of a given decision or its variation in response to any modification in the input data and/or in the DM preferences.
The aim of any decision model is to help the decider take decisions. The *final recommendation* in MCA may take different forms, according to the manner in which a problem is formulated. Roy (1985) identifies four objectives corresponding to four various problem formulations: choice, sorting, raking or description (see Table 1). Along with the nature of the decision problem, one formulation may be more efficient than an other. Lastly, we note that practical problems may require more than one formulation.

Problematic	*Objective*
Choice-oriented (P$_\alpha$)	Selecting a restricted set of alternatives
Sorting-oriented (P$_\beta$)	Assigning alternatives to different pre-defined categories
Ranking-oriented (P$_\gamma$)	Classifying alternatives from best to worst with eventually equal positions
Description-oriented (P$_\delta$)	Describing the alternatives and their follow-up results

Table 1: Decision-aid problematics

2.2 Multiple objective decision-making general model

The start point of any MODM technique is a set of *constraints* and a set of *objective functions*. The former set contains inequalities which reflect natural or artificial restrictions on the values of the input data. This means that feasible solutions are *implicitly* fixed in terms of these constraints.
In MODM, the DM preferences generally take the form of weights that are assigned to different objective functions. They may also be represented as *target values* that should be satisfied with any feasible solution.
The DM should also indicates, for each objective function, its sense of optimization, i.e., maximization or minimization. No other information than the weights and these senses of optimization are required to define the set of *non-dominated solutions*. This set contains solutions that are not *dominated* by any other one. The set of non-dominated solutions

is generally of a reduced size compared to the initial feasible solutions one. However, its generation usually requires a computer. This is explained by the high number of the initial feasible solutions to be evaluated.

Generally, *local and interactive aggregation algorithms* are used to define the feasible solutions set. This permits to combine the DM preferences and the computer to solve the decision problem, using methods that alternate calculation steps and dialogue steps. In reality, the local and interactive algorithms require the DM preferences to be expressed *progressively* during all the resolution process. The DM preferences, however, may be expressed a *priori* (i.e. before the resolution process) or *posteriori* (i.e. after the resolution process).

In many practical situations, the DM is called for to relax some of its constraints in order to guarantee that the set of feasible solutions is not empty or, simply, to test the stability of the results. Finally, we note that most of MODM problems are choice-oriented ones, aiming to find a 'best' solution (Vincke, 1992).

Table 2 below presents some corresponding points between the MADM and MODM general models.

MADM	*MODM*
Restricted set of alternatives	High or infinite number of feasible solutions
Explicitly defined set of alternatives	Implicitly defined set of feasible solutions
Different problem formulations	Choice-oriented formulation
Aggregation function is based upon an outranking relation or a utility function	Uses a local and an interactive aggregation algorithms
Requires a priori information on the DM's preferences	Requires much less a priori information on the DM's preferences

Table 2: Some characteristics of MADM and MODM general models

3 Extending the capabilities of the GIS

As it is underlined in the introduction, GIS technology is a limited tool in spatial decision-aid and an important question has emerged during the 1990s: *is the GIS a complete decision-aid tool*? Many recent works raise the crucial question of decision-aid within GIS (e.g. Janssen and Rietveld, 1990; Carver, 1991; Laaribi et *al.*, 1993, 1996; Malczewski, 1999). Most if not all of these works have come to the conclusion that GIS by itself can not be an efficient decision-aid tool and they have recommended the *marry* between GIS and the OR/MS and computing tools. The long-term objective of such an integration is to develop the so-called SDSS. The development of such a tool is an ambitious project that geos beyond the objective of this research. Instead, a design of a SDSS is proposed in the following section. Then, a more realistic solution is introduced in §3.2 and then implemented in §4.

3.1 Long-term solution: Developing a SDSS

What really makes the difference between a SDSS and a traditional DSS is the particular nature of the geographic data considered in different spatial problems. In addition, traditional DSSs are devoted almost only to solve structured and simple problems which make them non practicable for complex spatial problems. Since the end of the 1980s, several researchers have oriented their works towards the extension of traditional DSSs to SDSSs that support territory-related problems (Densham and Goodchild, 1989; Densham, 1991; Ryan, 1992; Chevallier, 1994; Malczewski, 1999). This requires the addition of a range of specific techniques and functionalities used especially to manage spatial data, to conventional DSSs. These additional capacities enable the SDSS to (Densham, 1991):

- acquire and manage the spatial data,
- represent the structure of geographical objects and their spatial relations,
- diffuse the results of the user queries and SDSS analysis according to different spatial forms including maps, graphs, etc., and to
- perform an effective spatial analysis by the use of specific techniques.

In spite of their power in handling the three first operations, GISs are particularly limited tools in the fourth one, which is relative to spatial analysis. Moreover, even if the GISs can be used in spatial problem definition, they fail to support the ultimate and most important phase of the general decision-making process relative to the selection of an appropriate alternative. To achieve this requirement, other evaluation techniques instead of optimization or cost-benefit analysis ones are needed. Undoubtedly, these evaluation techniques should be based on MCA.

As has already been indicated, the GIS-MCA integration constitutes an intermediate solution towards the development of a SDSS. Hence, to complete this integration successfully, it must be seen as an essential but not the only phase of a more general project aiming to build the so-called SDSS. A such project is beyond the scope of this research. However, a brief description of a design of a SDSS is provided hereafter.

The proposed design is conceived of in such a way that it supports GIS-MCA integration and is also open to incorporate any other OR/MS tool into the GIS. Figure 2 shows the components of the SDSS. These components are extensions of those of conventional DSSs which are here enriched with other elements required (i) to acquire, manage and store the spatially-referenced data, (ii) to perform the analysis of spatial problems, and

(iii) to provide to the decider and/or analysts an interactive, convivial and adequate environment for performing an effective visual decision-aid activity. These components are briefly depicted hereafter.

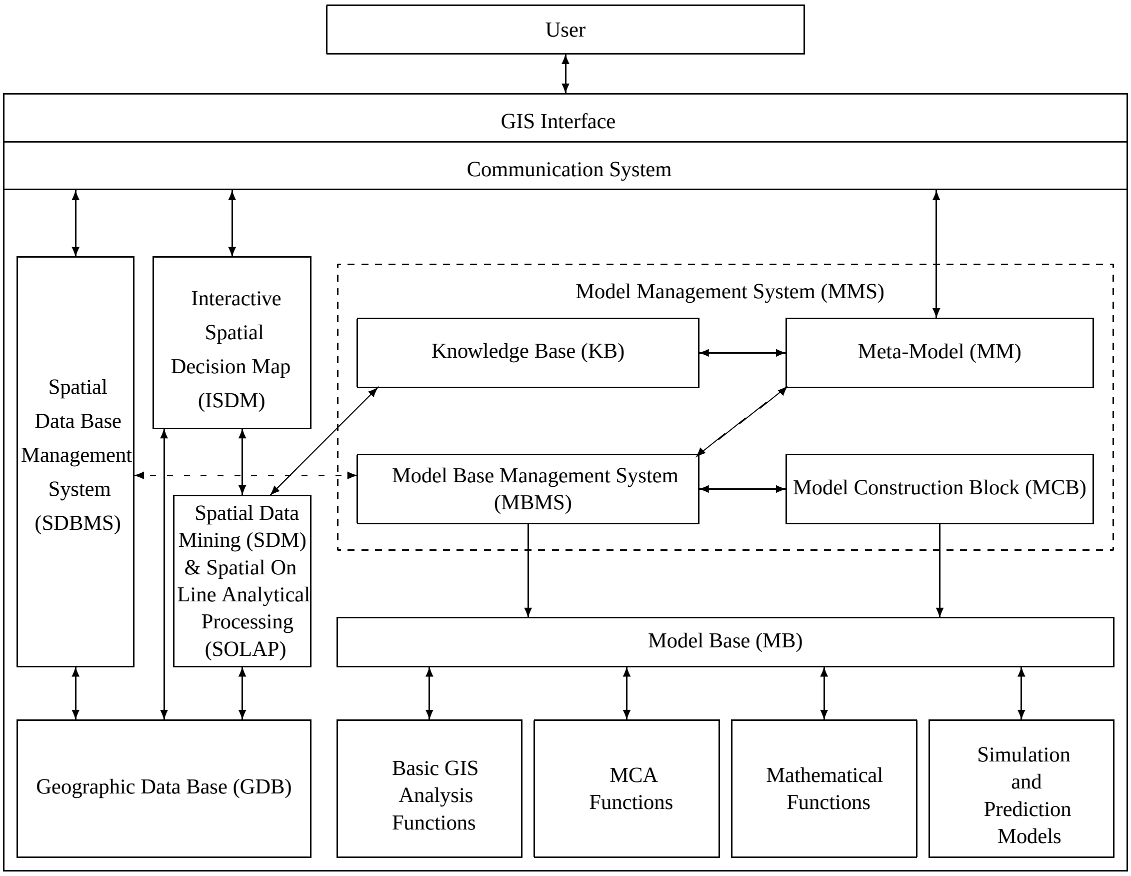

User
GIS Interface
Communication System
Spatial Data Base Management System (SDBMS)
Interactive Spatial Decision Map (ISDM)
Spatial Data Mining (SDM) & Spatial On Line Analytical Processing (SOLAP)
Model Management System (MMS)
Knowledge Base (KB)
Meta-Model (MM)
Model Base Management System (MBMS)
Model Construction Block (MCB)
Model Base (MB)
Geographic Data Base (GDB)
Basic GIS Analysis Functions
MCA Functions
Mathematical Functions
Simulation and Prediction Models

Figure 2: A design of a SDSS

Spatial Data Base Management System Spatial Data Base Management System (SDBMS) corresponds to the DBMS part of the GIS used specially to manage and store the spatial data. In fact, spatial data management is one of the powers of GISs.

Geographic Data Base The Geographic Data Base (GDB) is an extended GIS database. It constitutes the repository for both (i) the spatial and descriptive data, and (ii) the parameters required for the different OR/MS tools.

Model Base The Model Base (MB) is the repository of different analysis models and functions. Among these functions, there are surely the basic GIS ones (e.g. statistical analysis, overlapping, spatial interactions analysis, network analysis, etc.). This MB contains also other OR/MS models. Perhaps the most important ones are those of MCA. Nevertheless, the system is opened to include any other OR/MS tool (e.g. mathematical models, simulation and prediction models, etc.), or any other ad hoc model developed by the Model Construction Block (MCB) (see later in this section).

Model Management System The role of this component is to manage the different analysis models and functions. As it is shown in Figure 2, the Model Management System (MMS) contains four elements: the Meta-Model, the Model Base Management System (MBMS), the MCB and the Knowledge Base (KB).

Meta-Model The Meta-Model serves as a guide tool that helps the DM and/or analyst to select an appropriate model or function for the problem under study (Lévine and Pomerol, 1989, 1995) (see Figure 3). This element is normally an Expert System used by the DM to explore the MB. This exploration enables the DM to perform a 'what-if' analysis and/or to apply different analysis functions. The selection of the appropriate function depends on a base of rules and a base of facts continued in the KB.
Lévine and Pomerol (1995) advocate the fact that the Meta-Model is necessarily incomplete because a part of it resides in the decision-maker's mind. This enhances the role of the DM as an integral and indispensable component in the decision-making process.
Finally, we note that the notion of Meta-Model is of great importance in the sense that it makes the system open for the addition of any OR/MS analysis tool. This requires the addition of the characteristics of the analysis tool to the base rules, and, of course, the addition of this model to the MB.

Knowledge Base Knowledge Base is the repository for different pieces of knowledge used by the Meta-Model to explore the Model Base. Practically, the KB is divided into a base of facts and a base of rules. The base of facts contains the facts generated from the Model Base. It also contains other information concerning the uses of different models, the number and the problems to which each model is applied, etc. The base of rules contains different rules of decision which are obtained from different experts, or automatically derived, by the system, from past experiences. This base may, for instance, contains: *If the problem under study is the concern of many parties having different objective functions then the more appropriate tool is that of MCA.*

Model Base Management System The role of the Model Base Management System (MBMS) is to manage, execute and integrate different models that have been previously selected by the DM through the use of the Meta-Model.

Model Construction Block This component gives the user the possibility to develop different ad hoc analysis models for some specific problems. The developed ad hoc model is directly added to the Model Base and its characteristics are introduced into the base of rules of the KB.

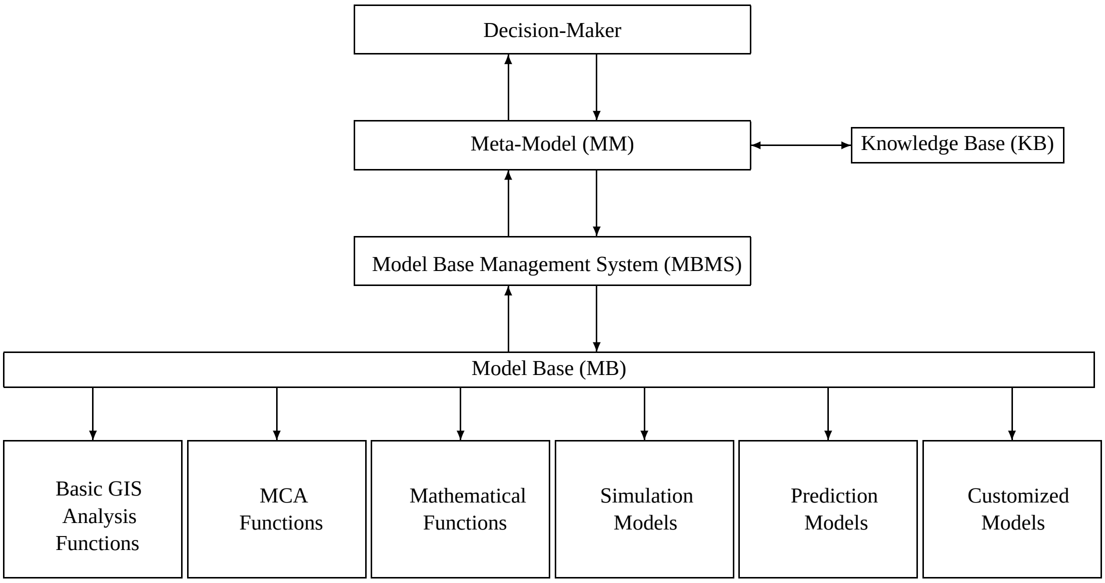


Figure 3: The role of the Meta-Model (inspired from Lévine and Pomerol (1989))

Spatial Data Mining and Spatial On Line Analytical Processing Data mining and On Line Analytical Processing (OLAP) have been used successfully to extract relevant knowledge from huge traditional databases. Recently, several authors have been interested in the extension of these tools in order to deal with huge and complex spatial databases. In particular, Faiz (2000) underlines that Spatial Data Mining (SDM) is a very demanding field that refers to the extraction of implicit knowledge and spatial relationships which are not explicitly stored in geographical databases. The same author adds that spatial OLAP technology uses multidimensional views of aggregated, pre-packaged and structured spatial data to give quick access to information. Incorporating SDM and SOLAP into the SDSS will undoubtedly ameliorate the quality of data and, consequently, add value to the decision-making process.

Interactive Spatial Decision Map Interactive Decision Map (IDM) is a new concept in decision-aid area (see, for e.g., Lotov et *al.*, 1997; Andrienko and Andrienko, 1999; Jankowski et *al.*, 2001; Lotov et *al.*, 2003). Its basic idea is to use map-based structures in order to provide an on-line visualization of the decision space, enabling the

decider-maker(s) to appreciate visually how the feasibility frontiers and criterion trade-offs evolve when one or several decision parameters change. The Interactive Spatial Decision Map (ISDM) component that we propose to integrate into the SDSS is an extension of this concept. It is an electronic map representing an advanced version of a classical geographic map, with which the decider-maker is quite accustomed, and which becomes a powerful visual spatial decision-aid tool, where the decider uses a representation very similar to real-world to dialogue with the database, explicit his/her preferences, manipulate spatial objects and modify their descriptive attributes, add/delete other spatial objects, simulate the adoption of a given scenario without alerting the original data, appreciate the effects of any modification affecting any preference parameter, order a decision, modify the decision space representation, use different spatial data exploration tools, etc.
It is important to note that the decision map does not subtract the utility of SDBMS. Rather, it constitutes another facet for managing spatially-referenced data. In fact, a SDBMS is exclusively oriented towards data management, while the decision map is oriented towards visual spatial decision-aid. They are two complementary tools requiring a maximum of coordination for their implementation.

Communication System The communication system represents the interface and the equipments used to achieve the dialogue between the user and the SDSS. It permits the DM to enter his/her queries and to retrieve the results.

As it is underlined above, the development of the proposed design is an ambitious project that require the collaboration of researchers from different disciplines and can be envisaged only in a long-term perspective. Instead, a more realistic solution that consists in the integration of GIS technology with MCA is provided in the following section.

3.2 Short-term solution: GIS and MCA integration

To avoid the limitations of GIS in spatial decision-aid, different researchers support the idea of integrating GISs with different OR/MS modelling and computing tools (e.g. Janssen and Rietveld, 1990; Carver, 1991; Fischer and Nijkamp, 1993; Laaribi et *al*., 1996; Malczewski, 1999; Laaribi, 2000). In fact, there are several concrete tentatives to integrate OR/MS and computing tools (such as linear programming, statistical analysis tools, neural networks, fuzzy sets, Expert Systems, etc.) and GIS in the literature. However, these first contributions ignore the multidimensional nature of territory-related problems. Equally, in these works the aspirations of decision-makers are completely neglected, which increases the gaps between GIS and SDSS. Moreover, in these works, modelling is usually achieved within an independent package and the GIS is served only as a visualization tool (Brown et *al*., 1994). In addition and as it is stated by Laaribi et *al*. (1996), in almost all these works the integration of GIS and OR/MS tools is achieved in a bottom-up way without any coherent framework. One possible path that has been supported by many authors and can be used to achieve the promise of GISs is to build a coherent theory of spatial analysis parallel to a theory of spatial data. Laaribi et *al*. (1996) believe that such an idea is foreseeable only in the long-term and they have proposed an intermediate solution consisting of integrating GIS with only a family of analysis methods. Perhaps the most convenient family is that of MCA. The role of the GIS in such an integrated system is to manage the spatially-referenced data associated with different spatial problems, while the role of MCA is to model these problems.

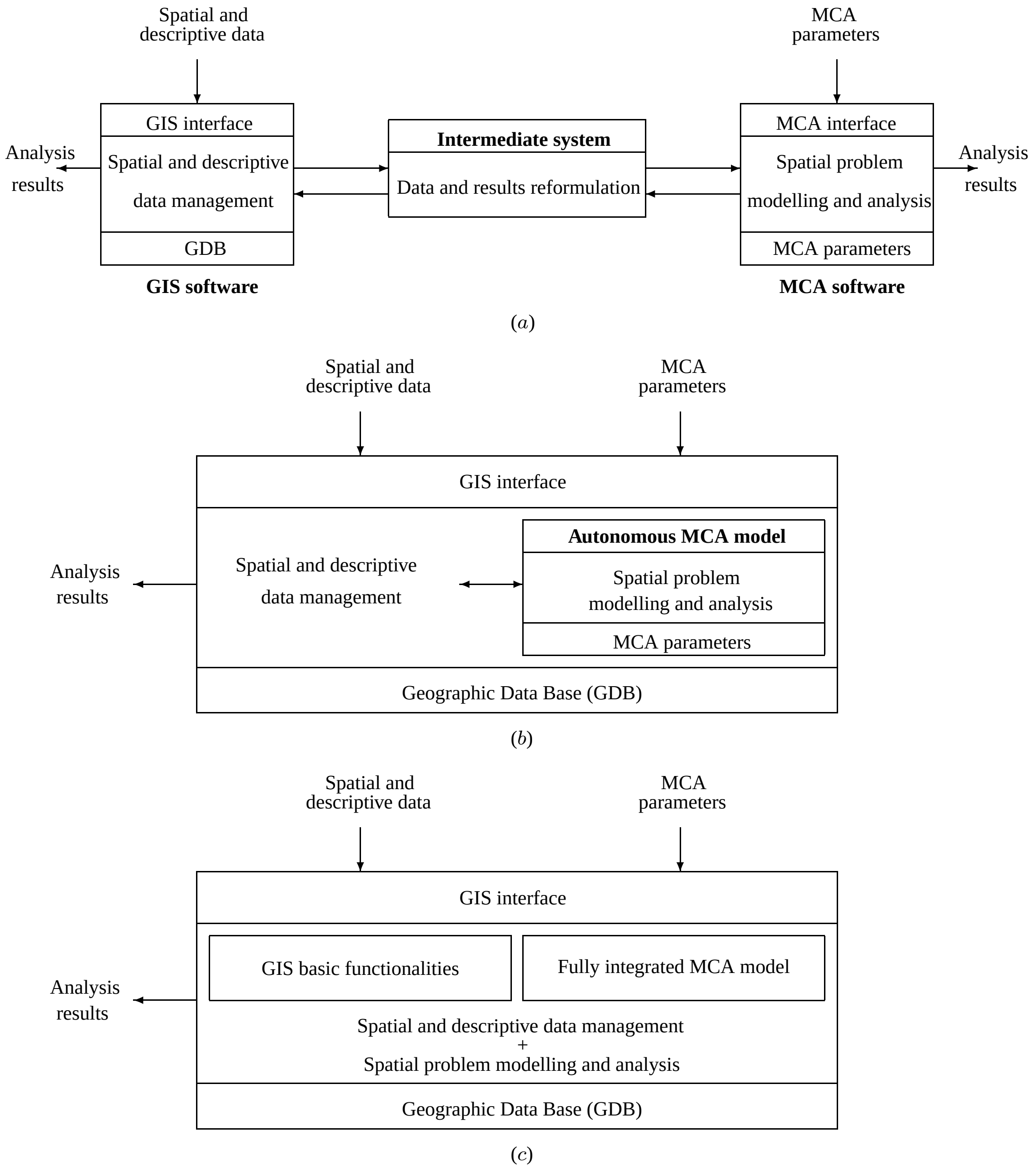
Several past works concerning GIS-MCA integration are avaliable in the literature (e.g. Janssen and Rietveld, 1990; Carver, 1991; Eastman et *al*., 1993; Pereira and Ducstein, 1993; Jankowski and Richard, 1994; Jankowski, 1995; Laaribi et *al*., 1996; Janssen and Herwijnen, 1998; Laaribi, 2000; Sharifi et *al*., 2002). What remain now is a consistent methodology for integrating efficiently the two tools.

The conceptual idea on which past works are based is the use of the GIS capabilities to prepare an adequate platform for using multicriteria models. Operationally, the GIS-MCA integrated system starts with the problem identification, where the capabilities of the GIS are used to define the set of feasible alternatives and the set of criteria. Then, the overlay analysis procedures are used in order to reduce an initially rich set of alternatives into a small number of alternatives which are easily evaluated by using a multicriteria model. Finally, the drawing and presenting capabilities of the GIS are used to present results.

Physically, there are three possible ways to integrate GIS and MCA tools. Our formulation of these three integration modes is illustrated in Figure 4 and described in the three following paragraphs. This formulation is inspired from the works of Goodchild (1991), Laaribi et *al*. (1993) and Jankowski (1995).

An indirect GIS-MCA integration mode The integration of a GIS software and a stand-alone MCA software is made possible by the use of an *intermediate system*. The intermediate system permits to reformulate and restructure the data obtained from the overlapping analysis which is performed through the GIS into a form that is convenient to the MCA software. The other parameters required for the analysis are introduced directly via the MCA software interface. The results of the analysis (totally made in the MCA part) may be visualized by using the presentation capabilities of the MCA package, or fedback to the GIS part, via the intermediate system, for display and, eventually, for further manipulation. It should be noted that each part has its own database and its own interface, which makes the interactivity a non-convivial operation.

A built-in GIS-MCA integration mode In this mode, a particular MCA model is directly added to the GIS software. The MCA model constitutes an integrated but autonomous part with its own database. The use of the interface of the GIS part alone


Spatial and descriptive data
MCA parameters
GIS interface
Spatial and descriptive data management
GDB
GIS software
Intermediate system
Data and results reformulation
MCA interface
Spatial problem modelling and analysis
MCA parameters
MCA software
Analysis results
Analysis results
Spatial and descriptive data
MCA parameters
GIS interface
Spatial and descriptive data management
Autonomous MCA model
Spatial problem modelling and analysis
MCA parameters
Geographic Data Base (GDB)
Analysis results
Spatial and descriptive data
MCA parameters
GIS interface
GIS basic functionalities
Fully integrated MCA model
Spatial and descriptive data management
+
Spatial problem modelling and analysis
Geographic Data Base (GDB)
Analysis results

(a)

(b)

(c)

Figure 4: The three GIS-MCA integration modes: (a) An indirect GIS-MCA integration mode, (b) A built-in GIS-MCA integration mode, and (c) A full GIS-MCA integration mode

increases the interactivity of the system. Here, there is no need for an intermediate system because the MCA model is reformulated in such a way that the exchange of data and analyzed results between the two parts is performed directly. This mode is the first step towards a complete GIS-MCA integrated system. Yet, with the autonomy of the MCA model, the interactivity remains a problem.

Full GIS-MCA integration mode The third mode yields itself to a complete GIS-MCA integrated system that has a unique interface and a unique database. Here, the MCA model is activated directly from the GIS interface as any GIS basic function. The GIS database is extended so as to support both the geographical and descriptive data, on the one hand, and the parameters required for the multicriteria evaluation techniques, on the other hand. The common graphical interface makes the global system a convivial single tool.

Clearly, the third way is the most efficient one for a powerful GIS-MCA integrated system. Consequently, it has been followed by several researchers (e.g. Jankowski, 1995; Laaribi et *al*., 1996). Their works have, however, a major limitation is that they integrate only one model (or a limited number of models), which makes the obtained system a rigid tool. Two possible ways to remedy this problem are (i) to integrate as many different MCA models as possible, or (ii) to integrate different multicriteria evaluation functions. Nevertheless, neither the first nor the second way would be able to solve the problem by itself. In fact, the first way generates a new problem relative to the selection of the appropriate MCA model in a given spatial problem, while the second way has no tool to choose the appropriate aggregation procedure. To take advantages of both ways and to avoid their respective limitations, an integration strategy is developed in the following section.

4 A strategy for GIS-MCA integration

The strategy developed in this section consists of the combination and extension of two past works. In fact, the idea of our strategy is to integrate (i) different multicriteria evaluation (MCE) functions, and (ii) a model for multicriteria aggregation procedure (MCAP) selection into the GIS. This new system uses a MCAP selection process of Laaribi et *al*. (1996), a general model of MADM proposed by Jankowski (1995) and illustrated in Figure 1(a), and the MODM general model illustrated in Figure 1(b).
Instead of integrating complete MCA models as was the case of almost all past works, this new strategy proposes the integration of only multicriteria evaluation functions. The idea is not totally new. It was proposed by Jankowski (1995) for incorporating MADM

common functions into a GIS. Nonetheless, this idea is here enriched with other functions and extended to include both MADM and MODM techniques.

4.1 Multicriteria evaluation functions

The MCA functions considered in the strategy are illustrated in Figure 1 and summed up in Table 3. The following paragraphs provide brief descriptions of the considered MCA common functions that are meant to be integrated into a GIS.

#	*Function*
F1	Alternatives\Criteria definition and generation
F2	Constraints definition
F3	Non-dominated solutions generation
F4	Objective functions definition
F5	Performance table generation
F6	Performance table normalization
F7	Weights assignment
F8	Aggregation procedure selection\definition
F9	Preferences definition
F10	Aggregation
F11	Sensitivity analysis
F12	Final recommendation

Table 3: MCA common functions

Alternatives\criteria definition and generation This function permits the user to explicitly define and then generate discrete sets of alternatives and criteria. There are two important remarks to arise at this level. First, the two sets are here defined jointly and not separately as in Jankowski (1995), which leads to a more realistic problem formulation. Second, the generation of the two sets should be normally handled by using the GIS overlay capabilities. However, the intervention of the DM and/or analyst for updating or may be a definition of both criteria and alternatives sets is a necessity. This is due to the fact that the overlays defined through the GIS are not true criteria. Rather, they are *admissibility criteria* or constraints. Nevertheless, a possibility of generating the two sets automatically by using the capabilities of the GIS in use is offered to the user via this function.

Constraints definition 'Constraints definition' function is used when the problem has very high or infinite number of feasible solutions, i.e., MODM problem. These constraints are used to generate the feasible solutions set. They can be defined explicitly by the DM or automatically through the overlapping capabilities of the GIS. In fact and as it is noticed in the preceding paragraph, the overlay analysis of GISs is a powerful tool to define admissibility criteria.

Non-dominated solutions generation Once this function is activated, it generates the non-dominated solutions set on the basis of the DM preferences represented in terms of target values, objective functions weights, etc.

Objective functions definition This function offers the user the possibility to define his/her objective functions in a context of a MODM problem. The user should define the sense of the optimization and the structure of each objective function. Of course, the variables used in the different functions are those used to define the set of constraints and consequently the set of feasible solutions.

Performance table generation The function ’performance table generation’ is required by almost all MADM techniques. Here, the DM is called for to articulate his/her preferences in terms of criteria scores.

Performance table normalization Criteria scores can be quantitative or qualitative and can be expressed according to different measurement scales (ordinal, interval, ratio). Thus, this function is used to re-scale (when it is necessary) the different criteria scores between 0 and 1. Different methods of normalization are offered to the user. The selection of the appropriate method depends greatly on the nature of the available data.

Weights assignment The criteria$\backslash$objective weights are a necessity for almost all MCA methods. The role of this function is to define a vector w of weights assigned to different criteria or objective functions. Several techniques for weighing criteria are available. The selection of the appropriate one depends heavily on the aggregation procedure used.

Aggregation procedure selection$\backslash$definition This function is the most important one in an integrated GIS-MCA system. It permits the DM and the analyst to choose the manner by which different criteria or objective functions are aggregated together. The selection of the appropriate aggregation technique is a very important step in MCA. In fact, there is a high number of methods; each of which has its advantages and disadvantages. In this sense, one method may be useful in some problems but not in others. The applicability of a given method depends on the aim of the problem formulation and on the data available, and the instant when these data are provided by the DM. Laaribi et *al.*(1996) propose a three-phases based model for selecting the more appropriate aggregation procedure. This model will be adopted in our strategy of integration and will be presented in more details later in this section.

Preferences definition As it is underlined in §2 and in addition to weights, the DM's preferences may also take the form of aspiration levels, cut-of values, or the form of target values. Contrary to weights which are required by almost all MCA models and to which we have associated a specific function, these ones (i.e. aspiration levels, cut-of values and target values) may or not be required in a given problem, this depends on the type of the aggregation procedure used. However, to make the system more flexible, this function will propose to the decider\analyst different possible ways for defining preferences and he\she should select the relevant ones. We note that in MODM problems, DM's preferences are usually expressed progressively during the resolution process. In this case, they are better defined through the 'Non-dominated solutions generation' function. Nevertheless, 'Preferences definition' function remains useful for MODM problems in which the preferences are expressed a *priori* or *posteriori* to the resolution process.

Aggregation As it is mentioned above, there are three families of aggregation procedures: (i) outranking relation-based family, (ii) utility function-based family, and (iii) local aggregation-based family. The two first ones are used in MADM problems and require that all the elements of the decision problem (i.e. alternatives, criteria, preferences) are defined. The third one is used in MODM problems within an interactive manner. Accordingly, aggregation for the methods of the two first families is handled through this function but local aggregation required by interactive methods is handled through the 'Non-dominated solutions generation' function.

Sensitivity analysis Nearly, all decision problems require a sensitivity analysis, enabling the DM to test the consistency of a given decision or its variation in response to any modification in the input data.

Final recommendation One of the basic concepts of decision-aid tools is that the final decision should never be taken by the system, i.e., the intervention of the DM in the selection and validation of a final recommendation is an inevitable phase (Lévine and Pomerol, 1995). Thus, the role of this function is only to provide the results of the analysis, via the capabilities of the GIS, to the DM to choose and, then, validate.

4.2 Aggregation procedure selection model

In Jankowski (1995), the selection of the appropriate aggregation procedure is guided only by the creativity of the DM (and/or analyst) and his/her knowledge in the domain of MCA, which may be very limited. A remedy to this consists in incorporating a model that will be used to guide the DM during the process of aggregation procedure selection. We note that the model that will be detailed hereafter is an implementation of the MCAP selection process that was proposed by Laaribi et *al*. (1996) and can be considered as a first version of the Meta-Model discussed in §3.1.

The principal idea beyond this model is stated as follows: the characteristics [in terms of the type of the problematic (choice, sorting, ranking or description), the nature of the set of alternatives (finite or infinite), the nature of the required information [measurement scale (ordinal, interval or ratio) and availability], and the type of the evaluation results [type of impacts (punctual or dispersed) and level of reliability]; see §6.2.3 in Laaribi (2000), pp.107-110)] of the spatially-referenced decision problem (SRDP) largely determine the characteristics of the MCAPs that are appropriate to the problem in question. Knowing the characteristics of the SRDP and the operating conditions of the MCAPs, one can identify the characteristics [in terms of the type of the problematic (choice, sorting or ranking), the nature of the set of alternatives (discrete or continuous), the nature of the required information [concerning criteria (ordinal or cardinal), and intra and intercriteria information], and the type of the evaluation results [type of result (punctual or dispersed) and the presence or absence of imprecision]; see Appendix 3 in Laaribi (2000), pp.162-164)] of the MCAPs that are suitable to the problem under focus.

Operationally, the model proceeds by elimination until the most convenient MCAP is identified. It starts with the identification of the characteristics of SRDP under focus. Then, these characteristics are used along with the operating conditions of the MCAP in order to identify the characteristics of the MCAPs which are appropriate to the problem in consideration. Once these characteristics are identified, they are superposed on a MCAPs classification tree (see Figure 5) to select a subset of suitable MCAPs, from which the DM can select the appropriate method. More formally, the steps of this model are (see Figure 6):

1. *Identification of the characteristics of the SRDP*. The decision-makers and/or analysts are called for to provide the characteristics of the SRDP.

2. *Identification of the characteristics of the MCAP*. The characteristics of the SRDP, along with the operating conditions of MCAPs, are used to identify the characteristics of the appropriate MCAP.

3. *Selection of a subset of MCAPs*. The characteristics of the appropriate MCAP are superposed on the classification tree in order to select a subset of suitable MCAPs.

4. *Selection of a MCAP*. Finally, the DM uses some intrinsic characteristics related to the use of MCAP to identify the most suitable MCAP.

This model, which is largely controlled by the DM, will guarantee him/her the selection of the most efficient aggregation procedure to the problem in consideration.
Two other simple models which can be used by the DM and/or analysts to respectively select the appropriate weighting technique and the normalization procedure, are added to the system. As it is underlined above, the selection of the appropriate weighting technique depends mainly on the nature of the aggregation procedure to be used, while the selection of the normalization procedure (if it is necessary), depends essentially on the nature of the available information.

4.3 Advantages of the strategy

The combination of GIS, MCE functions and MCAP selection model in a single tool yields a new system (Figure 7) that gathers the power of GIS in data management and presentation, the potentiality of MCA in spatial problems modelling, and the efficiency of the MCAP selection model for choosing the aggregation procedure.
The proposed strategy of integration gathers the advantages of both Jankowski (1995) and Laaribi et *al*. (1996) approaches. Perhaps the most important ones are:

- the achievement of a basic tenet of any SDSS which is *flexibility*, i.e., the capability of the system to be applied to a large spectrum of spatial problems. Using different and independent functions, the DM has the possibility to freely integrate different ingredients of MCA techniques in order to ’create’ a new and customized model adapted to the problem in question.
- the ability of the DM to interact constantly during all the decision-making process phases. In fact, each step in the resolution of any problem is directly controlled by the DM.
- the ability to carry out the different steps of the decision-making process in a ’non-sequential’ schema.
- the strategy guarantees the DM to select the most appropriate MCAP for the problem under study.

4.4 Illustrative example

To better appreciate the proposed strategy considering the following hypothetical facility location problem, which is illustrated in Figure 8. The company involved in this il-

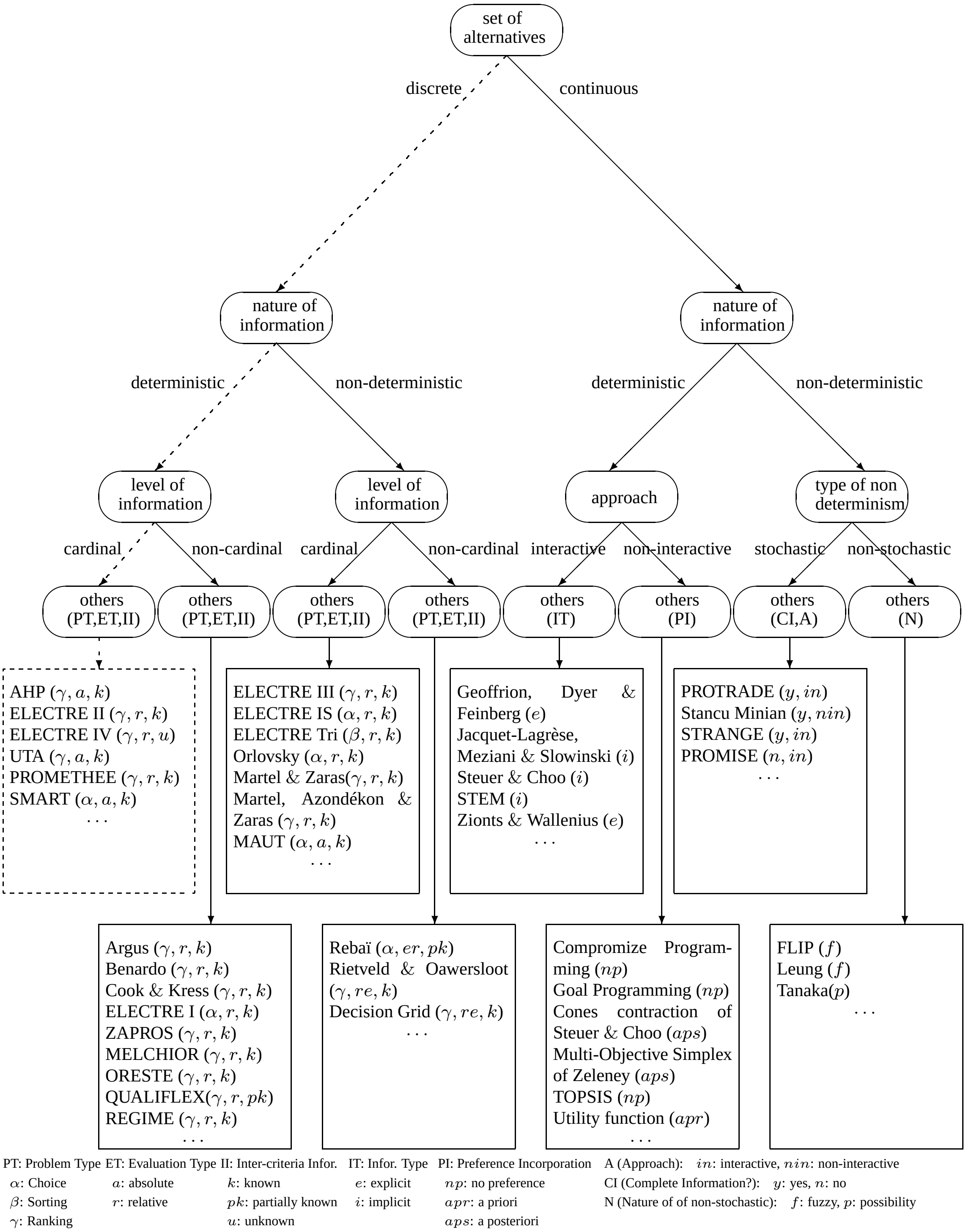


Figure 5: MCAPs classification tree [reproduced from Laaribi (2000), pp.114-115]

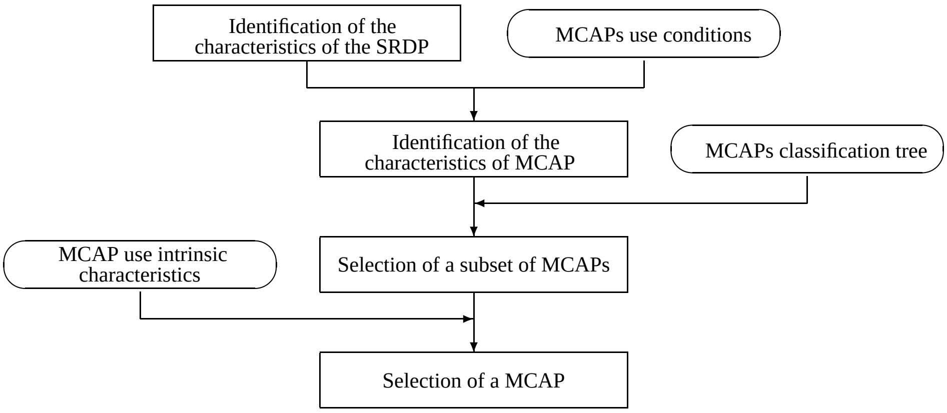

Identification of the characteristics of the SRDP
MCAPs use conditions
Identification of the characteristics of MCAP
MCAPs classification tree
MCAP use intrinsic characteristics
Selection of a subset of MCAPs
Selection of a MCAP

Figure 6: MCAP selection model

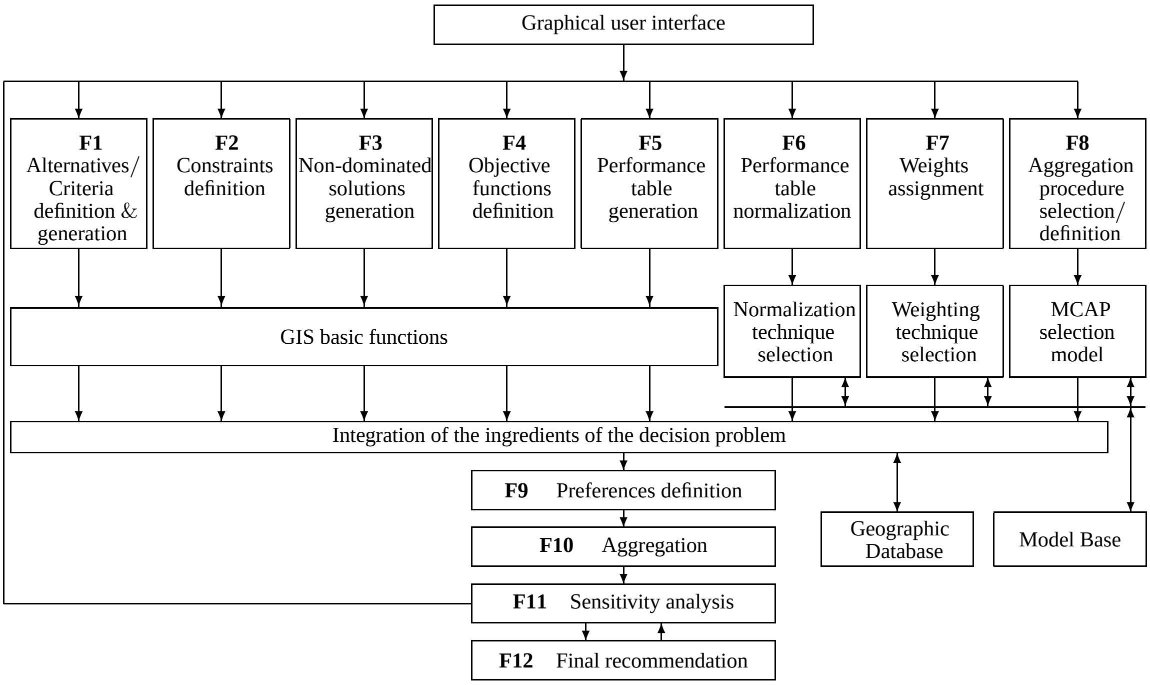

Graphical user interface
F1 Alternatives/ Criteria definition & generation
F2 Constraints definition
F3 Non-dominated solutions generation
F4 Objective functions definition
F5 Performance table generation
F6 Performance table normalization
F7 Weights assignment
F8 Aggregation procedure selection/ definition
GIS basic functions
Normalization technique selection
Weighting technique selection
MCAP selection model
Integration of the ingredients of the decision problem
F9 Preferences definition
F10 Aggregation
F11 Sensitivity analysis
F12 Final recommendation
Geographic Database
Model Base

Figure 7: Schematic representation of the strategy

lustrative example distributes petroleum products to different gas stations located throughout a given national market. The aim of this company is to determine the 'optimal' location for a new depot in order to avoid deliverance delay problems.
The responsible of the company has identified two siting factors which must be verified:

- the site must be within an area having a population greater than 50 thousand, and
- the new depot can not be located in an area where a depot already exits.

The first step is to use basic GIS functions to identify all the feasible sites. The inputs of this step is two layers representing the two factors cited above. These layers are then superposed upon each other and a logical addition operation is applied to them in order to identify areas that verify simultaneously the two factors.

In the next step one should apply function F1 ('Alternatives/Criteria definition and generation') to define the set of evaluation criteria. We suppose that the following evaluation criteria are considered in our example:

- distance from different pre-existing depots,
- number of potential customers in the rayon of 150 km,
- implementation cost, and
- proximity to routes

Once criteria are defined, the function F7 ('Weights assignment') is activated in order to define a vector of weights $w =< w_1, w_2, w_3, w_4 >$. In the next step, we may use function F5 ('Performance table generation') to calculate criteria scores for each feasible alternative. The first distance criterion can be easily performed through GIS basic functions. For each feasible location, the score of the second criterion can be obtained by summing the number of customers in all sites which are located no more than $150km$ from this location. The implementation cost criterion can be directly provided by the decider maker and/or analyst. The proximity criterion may be calculated as the number of routes which are no more than $100m$ far from the site.

Once all criteria scores are calculated, we may use function F8 ('Aggregation procedure selection/definition') to select an aggregation procedure. This function will activate the MCAP selection model in order to identify the most appropriate aggregation procedure(s). The progress of selection[2] is shown in dashed arrows in Figure 5, which indicates that the decider and/or analyst should select one procedure from the left-side dashed box of Figure 5. Then, we may apply function F10 ('Aggregation') in order to globally evaluate the different feasible alternatives.

Then, the result of aggregation and evaluation steps are visualized using the capabilities of the GIS. The decider and/or analyst may then use function F11 ('Sensitivity analysis') as many as necessary to see the effects of any modification on the input data. Finally function F12 ('Final recommendation') is used to adopt a given alternative.

[2] In this example, the set of alternatives is *discrete*, the nature of information is *deterministic* and the level of information is *cardinal*.

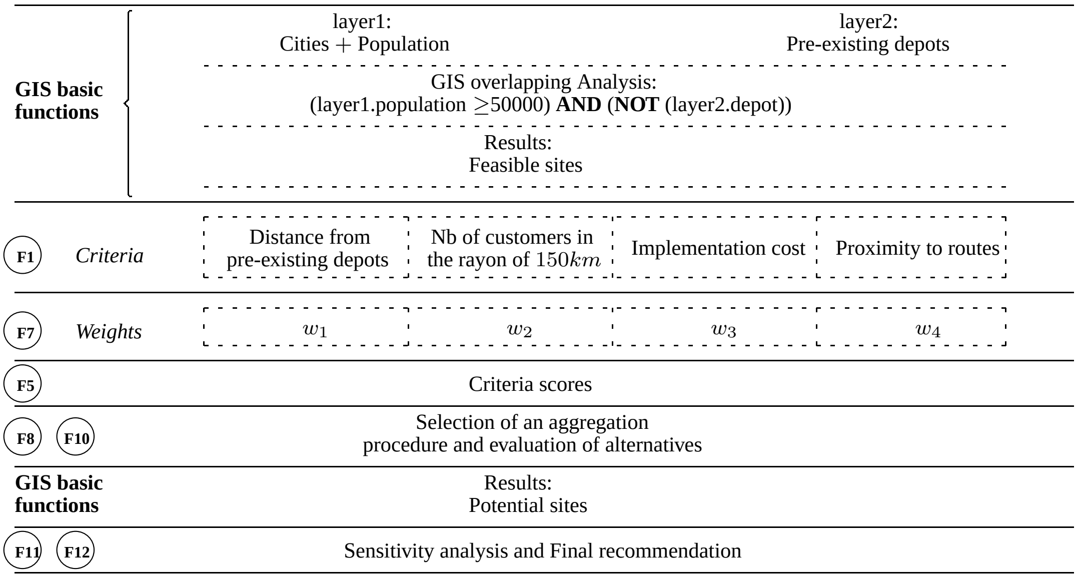

GIS basic functions
layer1:
Cities + Population
layer2:
Pre-existing depots
GIS overlapping Analysis:
(layer1.population ≥50000) AND (NOT (layer2.depot))
Results:
Feasible sites
F1
Criteria
Distance from pre-existing depots
Nb of customers in the rayon of 150km
Implementation cost
Proximity to routes
F7
Weights
w1
w2
w3
w4
F5
Criteria scores
F8
F10
Selection of an aggregation procedure and evaluation of alternatives
GIS basic functions
Results:
Potential sites
F11
F12
Sensitivity analysis and Final recommendation

Figure 8: Illustrative facility location example

5 Concluding remarks

In this paper we have proposed a strategy for integrating GIS and MCA. The idea of our strategy is to integrate (i) different MCE functions, and (ii) a Model for MCAP selection into the GIS. Depending on the two general models of MCA that are illustrated in section 2, twelve different MCE functions are isolated. These functions represent the different operations required to perform any MCA model. The separation of these functions permits to achieve an important characteristics of any SDSS which is flexibility. In reality, this will offer the DM the possibility to integrate different ingredients of MCA techniques in order to 'create' a new and customized model adapted to the problem in question. Moreover, this will lead the DM to interact constantly during all the decision-making process phases.

Given the fact that the MCAP is the most important ingredient of any MCA model, the selected procedure will largely influence the final recommendation. Thus, the strategy proposes the incorporation of a specific model to aid the DM to select the most appropriate procedure for the problem under focus.

The combination of GIS, MCE functions and MCAP selection model in a single tool yields a new system that gathers the power of GIS in data management and presentation, the potentiality of MCA in spatial problems modelling and the efficiency of the MCAP selection model to choose the aggregation procedure.

Currently, the proposed strategy is being implemented by using the GIS ArcView. An immediate perspective of our strategy is to add, to GIS, other OR/MS tools in addition to MCA. The first candidates are prediction tools including time series, forecasting techniques, and so on. This relies on the fact that spatial objects and phenomena are subject to numerous spatio-temporal changes. Thus, the predictions tools will serve to model these changes and will make the system a powerful planning tool. A long-term possible extension of this research consist in the conception and the implementation of the design of the SDSS detailed in §3.1.

Acknowledgement

The authors are really thankful to the anonym reviewer for his/her valuable comments on an earlier version of this paper.

References

[1] G. Andrienko and N. Andrienko. Interactive maps for visual data exploration. *International Journal of Geographical Information Science*, 13(4):355–374, 1999.

[2] V. Belton and T.J. Stewart. *Multiple Criteria Decision Analysis: An integrated approach*. Klumwer Academic Publishers, Boston, 2002.

[3] S. Brown, H. Schreier, and L. Vertinsky. Linking multiple accounts with GIS as decision support system to resolve forestry wildlife conflicts. *Journal of Environmental Management*, 42(4), 1994.

[4] P.H.A. Burrough. Methods of spatial analysis in GIS. *International Journal of Geographical Information Systems*, 4(3):221–223, 1990.

[5] P.H.A. Burrough and R.A. McDonnell. *Principles of Geographical Information Systems*. Oxford University Press, New York, 1998.

[6] S. Carver. Integrating multicriteria evaluation with GIS. *International Journal of Geographical Information Systems*, 5(3):321–339, 1991.

[7] J.-J Chevallier. De l'information à l'action : vers des systèmes d'aide à la décision à référence spatiale (SADRS). In *EGIS/MARI'94*, Paris, 1994.

[8] P.J. Densham. Spatial decision support systems. In D.J. Maguitre, M.F. Goodchild, and D. Rhind, editors, *Geographical Information Systems: Principles and Applications*, pages 403–412. Longman, London, 1991.

[9] P.J. Densham and M.F. Goodchild. Spatial decision support systems: A research agenda. In *Proceedings SIG/LIS'89*, volume 2, pages 707–716, Orlando, Florida, 1989.

[10] R.J. Eastman, P.A.K., Kyen, and J. Toledno. A procedure for multiple-objective decision making in GIS under conditions of conflicting objectives. In J. Hents, H.F.L., Ottens, and H.J. Scholten, editors, *Fourth European Conference on GIS (ESIG'93) Proceedings*, volume 1, pages 438–447. Genoa, Italy, 1993.

[11] S. Faiz. Managing geographic data quality during spatial data mining and spatial OLAP—Data warehousing and data quality. *GIM International*, 14(12):28–31, 2000.

[12] M.M. Fischer and P. Nijkamp. Design and use of geographic information systems and spatial models. In M.M. Fischer and P. Nijkamp, editors, *Geographic Information Systems, Spatial Modelling and Policy Evaluation*. 1993.

[13] M.F. Goodchild. Progress on the GIS research agenda. In J. Harts, H.F.L. Ottens, and M.J. Sholten, editors, *Proceedings of The Second European Conference on Geographical Information Systems (ESIG'91)*, volume 1, pages 342–350. Brussels, Belgium, April 2-5 1991.

[14] M.F. Goodchild. Geographical information science. *International Journal of Geographical Information Systems*, 6(1):31–45, 1992.

[15] C.L. Hwang and K.L. Yoon. *Multiple Attribute Decision Making: Methods and Applications*. Spring-Verlag, N.Y., 1981.

[16] P. Jankowski. Integrating geographical information systems and multiple criteria decision-making methods. *International Journal of Geographical Information Systems*, 9(3):251–273, 1995.

[17] P. Jankowski, N. Andrienko, and G. Andrienko. Map-centred exploratory approach to multiple criteria spatial decision making. *International Journal of Geographical Information Science*, 15(2):101–127, 2001.

[18] P. Jankowski and L. Richard. Integration of GIS-based suitability analysis and multicriteria evaluation in a spatial decision support system for route selection. *Environment and Planning B*, 21:323–340, 1994.

[19] R. Janssen and P. Rietveld. Multicriteria analysis and geographical information systems: An application to agriculture land-use in Netherlands. In H.J. Scholten and J.C.H. Stillwell, editors, *Geographical Information Systems for Urban and Regional Planning*, pages 129–139. Kluwer Academic Publishers, Dorchecht, 1990.

[20] R. Janssen and M. van Herwijnen. Map transormation and aggregation methods for spatial decision support. In E. Beinat and P. Nijkamp, editors, *Multicriteria Analysis for Land-Use Management*, Number 9 in Environment and Management Series, pages 253–270. Kluwer Academic Publishers, Dordrecht, 1998.

[21] A. Laaribi. *SIG et Analyse multicière*. Hermes Scienes Publications, Paris, 2000.

[22] A. Laaribi, J.J. Chevallier, and J.-M. Martel. Méthodologie d'intégration des SIG et de l'analyse multicritère. *Revue Internationale de Géomatique*, 3(4):415–435, 1993.

[23] A. Laaribi, J.J. Chevallier, and J.-M. Martel. A spatial decision aid: A multicriterion evaluation approach. *Computers and Urban Systems*, 20(6):351–366, 1996.

[24] A.V. Lotov, V.A. Bushenkov, A.V. Chernov, D.V. Gusev, and G.K. Kamenev. Internet, GIS, and interactive decision map. *Journal of Geographical Information and Decision Analysis*, (1):118–143, 1997.

[25] A.V. Lotov, G.K. Kamenev, and V.A. Bushenkov. Informing decision makers on feasibility frontiers and criterion tradeoffs by on-line visualization of Pareto frontier, 2003.

[26] P. Lévine and J.-Ch. Pomerol. *Systèmes interactifs d'aide à la décision et systèmes experts*. Hermès, Paris, 1989.

[27] P. Lévine and J.-Ch Pomerol. The role of the decision maker in DSSs and representation levels. In *Proceedings of the 28th Annual Hawaii International Conference on System Sciences*, pages 42–51. 1995.

[28] J. Malczewski. *GIS and mutlicriteria decision analysis*. John Wiley & Sons, New York, 1999.

[29] L.Y. Maystre, J. Pictet, and J. Simos. *Méthodes multicritères ELECTRE: Description, conseils pratiques et cas d'application à la gestion environnementale*. Presse Polytechniques et Universitaires Romandes, Lausanne, 1994.

[30] J.M.C. Pereira and L. Duckstein. A multiple criteria decision-making approach to GIS-based land suitability evaluation. *International Journal of Geographical Information Systems*, 7(5):407–424, 1993.

[31] J.-Ch. Pomerol and S. Barba-Romero. *Choix multicritère dans l'entreprise: Principe et pratique*. Hermès, Paris, 1993.

[32] T.C. Rayan. Spatial decision support systems. In *Proceedings URISA'92*, volume 3, pages 49–59, Washington DC, 1992.

[33] B. Roy. *Méthodologie Multicritère d'Aide à la Décision*. Economica, Paris, 1985.

[34] B. Roy and D. Bouyssou. *Aide multicritère à la décision: Méthodes et cas*. Economica, Paris, 1993.

[35] M.A. Sharifi, W. van den Toorn, A. Rico, and M. Emmanuel. Application of GIS and multicriteria evaluation in locating sustainable boundary between the Tunari National Park and Cochabamba City (Bolovia). *Journal of Multi-Criteria Decision Analysis*, 11(3):151–164, 2002.

[36] H.A. Simon. *The new science of management decisions*. Random House, New York and Evanston, 1960.

[37] Ph. Vincke. *Multicriteria Decision-Aid*. John Wiley and Sons, Chichester, U.K., 1992.

[38] S.H. Zanakis, A. Solomon, N. Wishart, and S. Dublish. Multi-attribute decision making: A simulation comparison of select methods. *European Journal of Operation Research*, (107):507–529, 1998.

Towards a Typology of Spatial Decision Problems[*]

Salem Chakhar[†], Vincent Mousseau[†]

Résumé

L'objectif de ce papier est de présenter une typologie des problèmes spatiaux. La typologie proposée est obtenue par un croisement entre les différents types d'entités spatiales associées aux actions potentielles (i.e. point, ligne, polygone ou réseau) et les différents modes qui peuvent être utilisés pour approcher les problèmes spatiaux (i.e. statique, temporel, temps réel ou séquentiel). La combinaison de ces deux dimensions donne exactement seize familles de problèmes spatiaux de base. La typologie obtenue n'intègre pas explicitement les problèmes impliquant des actions représentées par des entités composées. Néanmoins, elle reste adéquate pour décrire la plupart de ces problèmes du fait que ces derniers sont souvent décomposés en une série de problèmes de base impliquant chacun une seule action atomique.

Mots-clefs : Problèmes spatiaux, Actions potentielles, Typologie, Dynamic spatiale

Abstract

The aim of this paper is to establish a typology of spatial decision problems. The proposed typology is obtained by a crossover between the different types of spatial entities associated with spatial potential actions (i.e. point, line, polygon or network) and the different modes that can be used to approach spatial decision problems (i.e. static, temporal, real-time or sequential). The combination of these two dimensions provides exactly sixteen basic families of spatial decision problems. The obtained typology does not include explicitly problems that involve actions based on composed entities. Nevertheless, it is still adequate to describe most of them since these last ones are usually decomposed into a series of basic problems, each one involves only one atomic action type.

Key words : Spatial decision problems, Potential actions, Typology, Spatial dynamics

[*] An earlier version of this paper was presented at *The 58th Meeting of MCDA European Working Group, Moscow, Russia, October 9 to 11th, 2003.*

[†] LAMSADE, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16, France. {chakhar,mousseau}@lamsade.dauphine.fr

1 Introduction

Spatial decision problems may be roughly defined as those problems in which the decision implies the selection among several potential actions or alternatives[1] that are associated with some specific locations in space. Examples of spatial problems include: facility location, health care planning, forest management, vehicle routing, administrative redistricting, etc. In all these examples of spatial decision problems, the potential actions are characterized at least with their geographic positions and the selection of the appropriate one(s) will depend on the satisfaction of some space-dependent constraints and the 'optimization' of one or several space-related evaluation criteria.

Spatial decision problems are the concern of researchers from diverse disciplines (e.g. economists, planners, environmentalists and ecologists, politicians, scientists, etc.) that have different concepts and paradigms. Indeed, each of these researchers has a different perception and conception of real-world, which is in relation with its objectives and preoccupations. Thus, to improve the understanding of the aspects and the specificities of these problems and to establish an adequate framework for multidisciplinary researches, we think that the elaboration of a classification of spatial problems in a purely abstract form, devoid of any socio-economic, political, environmental, etc., contexts, is a good starting point.

The objective of this paper is thus to present a typology of spatial problems which is obtained by crossovering the different types of potential actions (as spatial entities) with the different modes that can be used to approach spatial decision problems. In fact, in spatial decision-making context, potential actions are usually assimilated to one atomic spatial entity[2] (i.e. point, line, polygon or network) or to a combination of several atomic spatial entities. Furthermore, spatial decision problems involve several spatial, natural or artificial, objects and phenomena, which have an inherent dynamic nature. In practice, however, this inherent dynamic nature of real-world may or not be taken into account. This depends on the nature of the spatial system to which the problem under consideration refers and equally on the objectives of decision-making. Hence, we may distinguish four ways for approaching spatial problems: static, temporal, real-time and sequential.

These ways of representing spatial potential actions and of approaching spatial decision problems are orthogonal and all combinations are possible. The combination of these two dimensions provides exactly sixteen basic families of spatial decision problems that will be detailed in §6. The obtained typology does not include explicitly problems that

[1] The two terms 'action' and 'alternative' are slightly different. In fact, the term 'alternative' applies when actions are mutually exclusive, i.e., selecting an action excludes any other one. In practice, however, we may be called to choose a combination of several actions (see, for e.g., example 5 in §6.3), which violates the exclusion hypothesis. In this paper, we adopt the term 'action' since it encompasses both 'action' and 'alternative' terms.

[2] In this paper, 'spatial entities' are conceptual abstractions of real-world objects, events and phenomena that are located in space or going on some locations in space (see §3).

involve actions based on composed entities. Nevertheless, the typology is still adequate to describe most of them since they are usually decomposed into a series of basic problems, each one involves only one atomic action type.

The rest of the paper is structured as follows. In section 2 we present some existing typologies. Section 3 is devoted to present some characteristics of spatial entities and to illustrate their inherent dynamic nature. Next in section 4, we distinguish the different possible ways for approaching spatial decision problems. Section 5 shows how spatial potential actions can be represented with spatial entities. Then, section 6 is devoted to present our typology. At this level, we first distinguish and detail two general typologies, and then a series of illustrative examples are provided in order to better appreciate the characteristics of eight different families of problems. Follows in section 7, we show how problems that involve complex actions can be represented and modelled as a series of atomic actions-based problems. Finally, section 8 concludes the paper.

2 Some existing typologies

An intuitive classification of spatial decision problems is the one based on the socio-economic and/or environmental domain to which the problem refers. We distinguish, for instance, facility location, land-use planning, service coverage, resource allocation, health care planning, vehicle routing, redistricting and forest management problems.

Spatial problems, as non spatial ones, can also be regrouped according to the quantity and the type of available information into (Leung, 1988; Munda, 1995; Malczewski, 1999): (a) deterministic problems (b) stochastic problems, and (c) fuzzy problems, which are respectively based on the use of perfect, probabilistic, and imprecise information.

Keller (1990) classifies spatial decision problems according to the number of criteria and the number of deciders that they involve. He identifies four classes: one class contains some problems that involve only one criterion and only one decision-maker, two classes containing some more problems which involve respectively only one criterion and several decision-makers or several criteria and only one decision-maker, and a class that contains problems that involve several criteria and several decision-makers. Keller (1990) points out that most of spatial decision problems belong to this last class.

Jankowski (2003) subdivides land-related decision problems into four common types: (a) *site* (or *location*) *selection*, concerned with the rank of a set of sites in priority order for a given activity (e.g. what site might best be for a particular type of business?), (b) *location allocation* concerned with stating a functional relationship between the attributes of the land and the goal(s) of the decision-maker(s) (e.g. where to locate a new fire station so that the least amount of the population has no more than 10 minutes response time?), (c) *land use selection* (or *alternative uses*) looking in ranking the uses for a given site in priority order (e.g. given a property, what can it be used for?), and (d) *land-use allocation* looking in defining the best uses for an array of sites (e.g. how much of the land should

be allocated for the following uses: forestry, recreation, and wildlife habitat?).
The first classification seems to be of a general interest. However, it tells nothing about the specificity of each family of problems and focalizes only on the socio-economic and/or environmental contexts to which problems belong. The two next typologies are not specific for spatial context. Besides, groups in both of them are large, making it difficult to extract out their common and general characteristics. Moreover, in the second classification, one problem can be assigned indifferently to the three groups according to the accuracy of the available data, which may evolve over time. Additionally, in real-world problems we usually make use of a mixture of deterministic, probabilistic and fuzzy data, which complicates the assignment of the problem under study to a given class. The fourth typology focuses only on land-use and consequently ignores a large spectrum of spatial problems. Finally, all typologies do not include explicitly the temporal dimension of spatial decision problems and ignore the specificities of spatial potential actions.

3 Spatial entities and their dynamics

Maguire et *al.* (1991) distinguish four families of spatial entities: (i) physical objects such as buildings, highways, etc., (ii) administrative units like communes, counties, parks, etc., (iii) geographic phenomena such as temperature, disease distribution, wind fields, etc., and (iv) derived information representing non-real phenomena such as ecological and environmental impacts of a nuclear central, suitability for cultivation, etc. Notice that in geographical information system (or GIS) community, geographic phenomena (or non-real phenomena) are considered as spatial entities when they are geographically located with their proper attributes (e.g. temperature or precipitation in Paris), with the geometric forms that represent them on maps (e.g. Severe Acute Respiratory Syndrome, or SARS, disease distribution in southern-east Asia may be represented with several polygons corresponding to the affected countries or zones), or with both of them (Bedard, 1991).
Whatever their natures, spatial entities have several characteristics that distinguish them from non spatial data. These characteristics are generally regrouped into several dimensions. Stefanakis and Sellis (1997) enumerate six dimensions along which spatial entities are defined. The ones that are relevant in this paper are: (a) *thematic* (or *descriptive*) *dimension*, which describes non spatial characteristics of entities (e.g. soil type, parcel number, color, PH, civil address, etc.), (b) *spatial dimension* that describes the spatial characteristics of geographic entities in terms of *position*, *geometry* and *topology*, and (c) *temporal dimension*, which describes the temporal characteristics of geographic entities in terms of *temporal position* that represents the occurrence (e.g. date of facility construction), or duration (e.g. period of land ownership) of spatial entities in or over time, *temporal behavior* that refers to the evolution of geographic entities in time, and *temporal topology* that describes the spatial and functional relationships between geographic entities induced by their temporal position or behavior.

The association of thematic and spatial dimensions with temporal dimension confers an inherent dynamic nature to spatial entities. This dynamic nature is the result of a series of changes (natural or not) that touch one or several thematic and/or spatial characteristics of geographic entities. Basing on the works of Claramunt et *al*. (1997) and Frank (1999), Lardon et *al*. (1999) distinguish three types of changes induced by time over spatial entities: (a) *thematic changes* that refer to the ones that affect the descriptive characteristics of geographic entities without modifying their existence and their spatial extensions (e.g. evolution of the population of a town), (b) *spatial changes* that refer to changes in the spatial characteristics (i.e. position, geometry and topology) of entities that modify their spatial extensions (e.g. successive extensions of urban fabric of an agglomeration) or result in the movement of these entities (e.g. movement of fire fronts), without alerting their existence, and (c) *identity changes* that refer to changes that modify the identity of geographic entities: entities may be divided, regrouped, combined, etc., (e.g. definition of new cadastral or new pasture unities).

Actually, spatial entities are subject to a mixture of changes. A forest fire front or a flock of animals moves and changes form, velocity, and direction. Yet, a plane in navigation or an ambulance in service moves without changing its form. In the management of a hydraulic system, it is generally the level of water that changes. In a forest management problem, changes may affect the form (as a result of the plantation of new zones, or disappearance of some other zones following, for example, an excessive exploitation or a severe drought), the topological relations (as a result of the construction of new routes), and/or the thematic characteristics (introduction of new animal or vegetal species). Nevertheless, in practice attention is generally limited to only some aspects of changes as some ones are not pertinent to the problem under focus and because some others are conducted in a very low rhythm in comparison with human life (e.g. movement of continents).

In literature, dynamics of real-world is essentially addressed in database-oriented contexts where the main objective is to represent and digitize spatial entities and their dynamics. However, here, this dynamic nature should be appreciated in terms of the spatiotemporal evolution of the consequences and impacts of spatial potential actions. Indeed, spatial potential actions are often represented with one or a combination of several atomic spatial entities, and the consequences and impacts of these actions are usually assimilated and measured via the thematic and spatial characteristics of the spatial entities that represent them. In addition, consequences and impacts of spatial potential actions are dispersed over space and in time, which means that thematic, spatial and identity changes that affect spatial entities apply also to these consequences and impacts.

4 Modes for approaching spatial decision problems

Along with the nature of the problem and the objectives of the study, spatial entities may be classified into *static* or *dynamic*. Static entities are those which have not time-varying characteristics (e.g. buildings, mountains). On the contrary, dynamic entities have at least one of their spatial or descriptive characteristics that vary in time (e.g. lacks, rivers, highways where traffic rate changes dynamically, etc.).

In several situations, the dynamic nature of real-world is considered to have no effects on the outcomes of the decision-making process. In practice, however, the high equity of spatial decisions and their long-term impacts on population and environment impose, to some extent, an explicit incorporation of the dynamic nature of real-world into problem formulation, modelling and resolution. Accordingly, we can distinguish two different perceptions of the decision environment:

- a *static perception* in which the inherent dynamic nature of geographic entities and of their functional and spatial interactions are not recognized because they have no significant effects on the achievement of the decision-making process, or because their handling is expensive and/or complicated, or
- a *dynamic perception* in which the evolutionary nature of the decision environment is explicitly integrated in the problem formulation and modelling.

An explicit integration of real-world dynamics requires the availability of perfect predictions of real-world changes. In some situations changes of real-world may be captured quite accurately through forecasting and projection of current trends. However, in several other practical situations, it is not possible (or difficult and/or expensive) to produce such predictions. The solution generally adopted in similar situations consists in a decomposition of the initial problem into several sub-problems, which are addressed sequentially in time. The idea of decomposing spatial problems into a series of sub-problems is also useful for addressing problems that involve decisions that their implementation is of high equity. The incorporation of the dynamic nature of real-world may also be imposed by the dynamic nature of the spatial system to which the decision problem refers.

Consequently, along with the nature of the spatial system to which the problem under consideration refers and the objectives of decision-making, and whether real-world dynamic nature is considered or not and the manner by which this dynamic nature is handled, spatial decision problems may be addressed according to one of four possible visions: static, temporal, real-time and sequential. The following paragraphs provide brief descriptions of these visions.

Static vision In static vision we consider that real-world is *stable* over time. Accordingly, this vision applies to problems with no or less equity and involves short or medium-term spatial decisions that have immediate consequences. In trip itinerary selection problem, for instance, decisions are generally tackled within ad hoc manner and the cost of a ’poor’ decision is simply put more time to arrive. Formally, in this vision the attention is focalized on a *unique decision*, which should be tackled basing on *punctual*

information that are available in the moment of making the decision and which represent a snapshot view of current (or predicted) real-world.
One possible way to take into account the dynamic nature of real-world while preserving a static vision is to make large enough all decision variables and parameters in order to be able to respond to any future evolution of real-world. This idea may be applicable in some simple situations. However, it will be unsuitable in several practical situations where changes are not linear. Moreover, a such practice may generate unnecessary expends if the parameters are over-evaluated. Thus, it is more interesting to approach spatial problems within a manner that integrates explicitly the dynamics of real-world. In practice this dynamic nature may be approached within a discrete or continues manners, which correspond respectively to temporal or real-time visions.

Temporal vision In temporal vision and contrary to real-time one, the time axis is approached within a discrete manner, i.e., we suppose that changes affecting real-world are conducted according to a low rhythm and that the essential of real-world dynamics can be resumed quite perfectly through time series-like functions. Formally, temporal vision focalizes on a *unique decision* which, contrary to the previous vision, should be tackled on the basis of *dispersed information* that represent predictions of real-world changes. Thus, this vision can be seen as an extension of the static one. It differs, however, by the strategic nature and the irreversibility of the decision to make, which impose a deeper understanding of this decision and demand the consideration of its future socio-economic, environmental and ecological impacts.
Practically, this formulation applies to problems with high equity that involve long-term spatial decisions. The problem of locating a nuclear central, for instance, involves strategic decisions which their consequences are dispersed over several dozens or may be hundreds of years. The negative effects of a nuclear central are nearly inevitable and the problem comes down to the minimization of long-term damages. This requires an explicit consideration of real-world dynamics—in terms of population growth, climatic changes, soil dynamics, future environmental impacts, for instance—during the decision-making process.

Real-time vision Contrary to temporal vision, in this one we consider that real-world is continuously changing and that changes affecting this real-world are often unpredictable. In practice, this vision applies to problems where: (a) a *series of decisions* should be tackled over time to reach one or several global objectives, (b) these decisions are *interdependent*, and (c) the decision environment is *dynamic*; it is subject to several changes that may result from natural phenomena and/or induced by decision-maker's previous decisions. Formally, this vision focalizes on a *series of decisions* which should be tackled under *time pressure* and basing on *instantaneous information* representing the state of real-world at the moment of decision-making.

In a fire fighting problem, for instance, fires start spontaneously (or accidently) and evolve in different directions as a response to previous decisions as well as to several exogen factors which the decider can not control as wind direction and velocity, temperature, forest density and type, etc. In such a context, decisions are taken under time pressure and the *timing* of these decisions will have major effects on the achievement of global objectives. At this level, it is important to notice that even though in temporal vision the dynamic aspects of real-world are explicitly integrated in the problem modelling, the timing of the decision to make is not relevant for the achievement of objectives. This is because the main aim of temporal formulation (as it is defined here) is to take into account the dynamics of real-world in such a way that the decision performed ’now’ remains ’optimal’ in long-term. In terms of a cost-benefit analysis, this means that it generates the ’best’ cumulated gain and the ’least’ cumulated negative effects.

Sequential vision Spatial problems may also be handled sequentially in time, where the initial problem is decomposed into a series of related static sub-problems. Formally, we assist to a *series of decisions* dispersed over time, each one is tackled basing on *punctual information* representing the state of real-world during the considered period of time. In practice, these decisions may be performed in different points of time—that correspond usually to the beginning of different planning periods—or defined simultaneously at the beginning of the first period as an *action plan*. In both cases, the time dimension intervenes implicitly in the problem modelling and resolution because it is not formally expressed in the problem formulation but it can be deduced.

In spatial context, it is usually the high level of uncertainty or the considerable financial, economic and human requirements that makes deciders behave as sequential decision-makers. In the first case, a sequential formulation permits to take into account impacts of pervious decisions and consequently to reduce (partially) the effects of uncertainty through a sequential information-acquisition process, while in the second case a such formulation permits particularly to subdivide the expends over several budgetary plans.

Static, sequential and temporal visions are more suitable to Simon’s (1960) decision-making process phases (i.e. *intelligence*, *design* and *choice* or *selection*) because they represent situations where "we have time to act". On the contrary, real-time vision requires that decisions are made under time pressure. Hence, it is concerned mainly with choice phase rather than intelligence or design ones.

On the other hand, these visions respond to different objectives. Selecting a particular vision depends on both the nature of the problem and the objectives of decision-making. Static and temporal ones occur generally in decision-aid perspectives. In static situation we suppose that the decision environment is stable, minimizing hence its effects on the decision-making process. In turn, in temporal vision, the dynamic nature of decision environment is explicitly integrated in the problem modelling and resolution. Consequently,

static and temporal visions respond respectively to short-term and long-term decision-aid perspectives. Sequential vision applies to spatial context essentially when spatial management and planning problems are seen as pure investment ones in which financial aspects are the more relevant elements for the decision-maker(s). In this case, the elaboration of a sequential decision logic for handling the problem permits to reduce progressively the uncertainty, which will ameliorate the achievement of the decision-makers' objectives. The dispersed nature of sequential decisions does not means that dynamic aspects of real-world are put in consideration. In fact, each sub-problem represents a static decision situation but the fact that these problems are resolved successively implies that the decision-maker takes the new decision after appreciating the consequences of pervious ones. Accordingly, sequential formulation of (spatial) decision problems are mainly interested in the 'sequential search for information to be used in the decision-making process' (Diederich, 1999). Unlike the previous situations, real-time one manifests essentially in a problematic of control and tracking of dynamic spatial systems and/or of moving objects and/or phenomena and, contrary to the temporal vision, operates in continually changing environment.
Finally, it is important to notice that these visions of spatial dynamics, whose their characteristics are summed up in Table 1, may apply also to non spatial decision problems and are not always crisply defined.

Vision	*Static*	*Temporal*	*Real-time*	*Sequential*
Decision environment	Stable	Dynamic	Dynamic	Stable
Nature of information	Punctual	Dispersed	Instantaneous	Punctual
Type of decision	Unique decision	Unique decision	Series of decisions	Series of decisions
Objective	Short-term decision-aid	Long-term decision-aid	Control of dynamic systems	Sequential search for information

Table 1: Characteristics of the different visions of spatial decision problems

5 Representing spatial potential actions

Spatial potential actions are defined with at least two elements (Malczewski, 1999): (a) *action* (what to do?) and (b) *location* (where to do it?). In real-time decision situations, a third element is required to define spatial potential actions: (c) *time* (when to do it?). As it is signaled above, even though that in temporal situation the temporal dimension is explicitly integrated into the problem formulation and modelling, attention is essentially focalized on one decision and the time when this decision is performed is *a priori* with no importance. The same remark holds in sequential situation because the timing of the different decisions is not relevant for the problem formulation. Indeed, the different temporal points correspond to the logic succession of these decisions.

In each (spatial) decision problem, we assign to each potential action one or several decision variables, permitting to measure the performance of this action. These variables may be binary, discrete or continuous. Illustrating this with some examples inspired from Malczewski (1999). In a nuclear waste deposit location problem, for instance, the decision "locate the deposit at site x" is an action geographically located (via the site address, for example). The binary variable associated to each potential action (site) is the binary decision "construct the deposit at site x" or "not construct the deposit at site x". In a school location problem, we may be concerned with the size of the school in terms of the number of students to be affected to it. So, to each potential action and in addition to the binary locational variable, we assign a discrete variable which determines the size of the school. If we return to the nuclear waste deposit location problem, one may also be called to use a new continuous variable to measure the deposit area.

On the other hand, in (multicriteria) spatial decision-aid activity, we generally represent potential actions through one of four atomic spatial entities, namely *point*, *line*, *polygon* or *network*. Therefore, in a facility location problem, potential actions take the form of points representing different potential sites; in a linear infrastructure planning problem (e.g. highway construction), potential actions take the form of lines representing different possible routes; and in the problem of identification and planning of a new industrial zone, potential actions are assimilated to a set of polygons representing different candidate zones (see Table 2).

Even though a ’network’ entity is a composed one, it is introduced here as an atomic primitive in order to handel some applications in which attention is focalized on the identification of some distribution networks. In a petroleum distribution problem, for instance, networks always refer to different policies of distribution, where nodes represent demand points and arcs represent routes between these demand points. Networks define different system of routes, each one is characterized with its level of coverage, transportation cost, deliverance rapidity and so on. The same remark holds for public services distribution (e.g. electricity and heat) where networks are the different possible spatial configurations, which differ with their implementation costs, coverage levels, their responses to congestion and saturation situations, etc.

Potential Action	*Typical problem*
Point	Site selection: points represent different potential sites
Line	Highway layout identification: lines represent possible routes
Polygon	Evaluation of construction zones: polygons represent different zones
Network	Goods distribution: networks are the different distribution policies

Table 2: Some examples of atomic spatial potential actions

The association between spatial entities and spatial potential actions means that these last ones have all the characteristics and the dimensions of the first, and that are subject to all the changes mentioned in §3. In fact and as it is underlined above, the consequences and impacts of spatial potential actions are often assimilated to the descriptive and spatial characteristics of spatial entities used to represent them. Accordingly, the performances of these potential actions according to different decision variables can easily be measured in terms of the descriptive and spatial characteristics that represent the consequences and impacts of these actions.

6 Proposed typology

One way to classify spatial decision problems is the one based on the type of the potential actions that they imply. Accordingly, we may distinguish four basic families of spatial decision problems, which correspond to the four atomic types of actions. This classification is mainly useful to define the types of operators and spatial routines susceptible to be used in the evaluation and comparison of potential actions. However, a such classification ignores the inherent dynamic nature of spatial problems. Earlier, we have seen that depending on the nature of the problem and the objectives of the study, a spatial problem may be formulated as a static, temporal, real-time or a sequential decision problem. Each of these visions requires a different form of data and calls for different modelling and resolution techniques. It follows thus that the adoption of a specific vision will have major effects on the problem modelling and resolution. Thus, to improve the understanding of the aspects and the specificities of spatial problems and to select the adequate modelling and resolution techniques, and to better define the spatial operators and routines susceptible to be used, a typology of spatial decision problems is detailed in the following paragraphs. The proposed typology is based on a crossovering of the different types of actions with the ways that can be used to approach spatial decision problems. It is summed up in Table 3. Two general classifications can be distinguished in this table: problem formulation and potential actions-oriented typologies.

6.1 Problem formulation-oriented typology

The first general typology contains four major families of spatial problems, which map to the four possible ways for approaching spatial decision problems. In the following paragraphs we will focus only on the modelling and resolution techniques and data structures required for each family of problems. Notice that the following descriptions apply to all types of potential actions; their presentation here will avoid redundancy.

	Static	*Temporal*	*Real-time*	*Sequential*
Point	Punctual actions-based spatio-static decision problems	Punctual actions-based spatio-temporal decision problems	Punctual actions-based spatio-real-time decision problems	Punctual actions-based spatio-sequential decision problems
Line	Linear actions-based spatio-static decision problems	Linear actions-based spatio-temporal decision problems	Linear actions-based spatio-real-time decision problems	Linear actions-based spatio-sequential decision problems
Polygon	Polygonal actions-based spatio-static decision problems	Polygonal actions-based spatio-temporal decision problems	Polygonal actions-based spatio-real-time decision problems	Polygonal actions-based spatio-sequential decision problems
Network	Network actions-based spatio-static decision problems	Network actions-based spatio-temporal decision problems	Network actions-based spatio-real-time decision problems	Network actions-based spatio-sequential decision problems

Table 3: The proposed typology

6.1.1 Spatio-static decision problems

This family regroups problems with less equity, where the dynamic nature of real-world is ignored. Techniques used to resolve spatio-static problems are fundamentally static because they do not integrate explicitly the dynamic aspects of real-world. Examples of techniques[3] include linear programming, multicriteria analysis, network analysis models, simulated annealing, neural networks, graph theory, multi-agents systems, genetic algorithms, flow analysis, etc. In all cases, evaluation and comparison of potential actions are based on punctual information issued from direct and punctual (in time) measurements of actions' attributes. These punctual information may also be issued from spatial and/or temporal total aggregations of dispersed data (e.g. average annual precipitation in a given region), an extrapolation of past data in the present time or a projection of current trends of real-world in the future. These information can be supported easily by conventional spatial data management systems (e.g. GIS).

6.1.2 Spatio-temporal decision problems

This family regroups problems with high equity where the dynamic nature of real-world is explicitly integrated in the problem modelling and resolution. This requires

[3] It is important to notice that even that these techniques do not include explicitly the dynamic aspects of spatial decision problems, they, however, may be extended to capture these aspects and be very useful in many non-static decision problems.

anticipations of future facts and events. Several techniques can be combined with static models in order to predict future and integrate effects of natural, social, economic, etc., transformations in modelling spatial problems as, for instance, those based on probabilistic representations or those that use belief functions or fuzzy sets. However, these techniques are based on a total temporal aggregation, which generates problems of temporal compensation. Some more elaborated techniques are also available: animation (morphing) techniques, spatio-temporal Makovian models, time series, regression equations, etc. The explicit integration of time dimension in spatial decision-aid context requires the use of dispersed and evolutionary information that permit to retrace spatial and temporal evolution of spatial entities. These dispersed information may take the form of a series of time-indexed values (e.g. population of a town taken at different dates, mensual precipitations of a given region, etc.) or the form of a discrete function (e.g. representing population evolution of a town with a function $p(t)$, $t \in \mathbb{Z}$). In both cases, a temporal spatial data management system (e.g. temporal GIS) is necessary for handling evolution of facts and events over space and in time.

6.1.3 Spatio-real-time decision problems

This family regroups problems related to the control of dynamic systems or to the tracking of moving objects or phenomena. In dynamic system-related problems, we usually consider that geographic position does not vary over time while at least one of the other characteristics is time-varying. In these problems we are interested in the study of trajectories followed by spatial entities in order to analyze their evolution and behavior. In problems of tracking of moving objects or phenomena, the geographic position is time-varying; the other characteristics may or not vary over time. In both cases, we consider that real-world changes continuously. Equally in both cases, decisions need to be made under time pressure, especially in tracking problems where the *timing* of decisions has important effects on the achievement of objectives. Techniques used to handel spatio-real-time problems include dynamic programming, multiobjective dynamic optimization, differential equations, cellular automata, simulation techniques, system dynamics, etc. In real-time situation, the required information evolve continuously and are often represented through continuous functions (e.g. representing movement of spatial engine with function $m(x, y, z, t)$, $t \in \mathbb{R}$, which gives positions (x, y, z) taken by the engine in different time instants t). A better handling of this kind of information necessitates the development of real-time spatial data management systems.

6.1.4 Spatio-sequential decision problems

Most of sequential problems are not spatial ones. However, several spatial problems may be formulated as sequential decision ones (especially by risk-averse deciders) mainly when there is a high level of uncertainty or when the decisions that they involve are of high equity. Tools as strategic choice approach and robustness analysis are often used to deal with non spatial problems characterized by a significant level of uncertainty. These tools may apply also to spatial problems essentially when only financial aspects of these problems are considered. However, they are of limited use when a variety of different social and environmental criteria should be included in the study. In addition, the two tools are more graphical technologies rather than formal mathematical formulations. There are many other more formal tools based on solid mathematical formulations such as Markovian decision models, dynamic programming tools, discounted utility-based models, etc. Nevertheless, most of these tools do not exploit the spatial characteristics of the problem components because they are initially conceived and used for non-spatial decision problems and they focalize only on the financial aspects of the problem. As in the first family, this one does not consider the dynamic nature of real-world and evaluation and comparison of potential actions are based on punctual information. However, in this case these information are often characterized with a high level of uncertainty and/or fuzziness, which require tools that are able to handel uncertain and fuzzy spatial data.

6.2 Potential actions-oriented typology

The second general typology is potential actions-oriented one. It is detailed hereafter. It contains four major families, which map to the four types of potential actions. Each of these families contains four sub-families which correspond to the four modes for approaching spatial decision problems. The following four paragraphs provide brief descriptions of these families. Then, in §6.3, we provide eight illustrative examples where the characteristics of eight sub-families of spatial problems are depicted.

6.2.1 Punctual actions-based spatial decision problems

Punctual entities are usually used to represent potential sites in location-related problems, particularly when only geographic position is considered. Punctual actions-based spatial decision problems may be further decomposed into four sub-families:

1. Punctual actions-based spatio-static decision problems

2. Punctual actions-based spatio-temporal decision problems

3. Punctual actions-based spatio-real-time decision problems

4. Punctual actions-based spatio-sequential decision problems

The first sub-family contains location problems in which the dynamic nature of the decision environment is neglected. Actually, most of practical location applications are handled as static decision problems. This may be true for facilities with no or less equity and with no or limited impacts on population and environment. However, facilities like airports, nuclear centers, hypermarkets, hospitals, universities, stadiums, and so on, have inevitably long-range impacts and depend on several socio-economic, environmental, ecological and political criteria which evolve over time.

As we have signaled above, spatial problems are formulated as sequential decision problems when there is a high level of uncertainty and/or when they are of high equity. Accordingly, some strategic location problems are approached sequentially in time. Considering the example of a multinational company which looks to open three new car assembling factories. Due to the fact that this investment project is of high equity and due to the high level of uncertainty (which may be related to demand, competition, foreign governments policies, etc.) that characterizes it, the responsible(s) may adopt a sequential decision-making strategy, where three-period planning horizon is defined. The original problem is thus decomposed into three inter-related static location decision problems, one for each period. The specificity of this situation, compared with the situation where a series of three non-related static problems are considered, is that the objectives of the three sub-problems are the same: minimize total implantation costs and maximize total coverage.

Punctual actions may also be useful in problems related to the control and tracking of dynamic or moving spatial punctual objects such as controlling and guiding ambulances and fire-fighter vehicles, military navies or aircrafts in navigation, etc.

As it is signaled above, the nature of the spatial entity by which decision actions are represented will determine largely the type of spatial and temporal operators and analysis routines susceptible to be used for evaluating and comparing these actions. Concerning punctual actions, all distance-based measurements (e.g. spatial and temporal distances, proximity) and several statistical and spatial operations such as buffering and interpolation are usually applied. Buffering operation and statistical analysis procedures are not specific for punctual actions and may apply to all types of actions.

Even though that the different sub-families cited above have the same nature (location), they have different objectives and require different data structures and different modelling and resolution tools. These differences have crucial impacts on the development of spatial decision-aid tools and their understanding will have good results on the efficiency of these tools. All these elements legitimacy, to some extent, the subdivision of punctual location problems into different sub-families. This is true also for families of problems based on other types of actions.

6.2.2 Linear actions-based spatial decision problems

Linear objects may represent several types of real-world decision actions such as highways and routes, rivers, gasolines, etc. Equally, problems of this category can be further decomposed into four sub-families:

1. Linear actions-based spatio-static decision problems
2. Linear actions-based spatio-temporal decision problems
3. Linear actions-based spatio-real-time decision problems
4. Linear actions-based spatio-sequential decision problems

Selecting an itinerary for making a trip involves a non strategic decision that has limited and immediate consequences, and can adequately approached via a static formulation. In turn, a highway construction problem involves a strategic decision that have long-term impacts on population, environment, ecology etc., and a temporal formulation will be more adequate. The static and temporal formulations differ essentially on the information on which the decision is taken: in the first case, we suppose that the selection of a particular itinerary has no long-term impacts, and static information are supposed to be sufficient for representing these impacts, while constructing a highway has long-term impacts which need to be explicitly incorporated in the problem modelling and resolution via the use of time-dispersed spatial information that reflect future evolution of these impacts.

Sequential formulation intervenes often when the linear planning problem is of high equity. In a such situation, the solution is to subdivide the original problem into several parts, which will be constructed in different dates.

Finally, real-time formulation manifests, for example, in the management of linear hydraulic systems (e.g. river) or highways traffic regulation, in shortest path problems for moving objects, etc.

In addition to statistical and buffering operations, the most used operations for linear actions are travel time and length measurements, and PointInLine operation, which tests the intersection of a point (e.g. bus station) with a line (e.g. trip itinerary).

6.2.3 Polygonal actions-based spatial decision problems

Polygons, which describe topological proprieties of areas in terms of their shapes, neighbors and hierarchy, are often used in regional planning and land-use-related problems, which Jankowski (2003) subdivides into four types (see §2):

- location selection problems concerned with the rank of a set of sites (i.e. polygons) in priority order for a given activity,
- location allocation problems concerned with stating a functional relationship between the attributes of the land and the goal(s) of the decision-maker(s),

- land-use selection problems looking in ranking the uses for a given site in priority order, and
- land-use allocation problems looking in defining the best uses for an array of sites.

In the first type of problems, polygons represent candidate areas for a given industrial, commercial or social activity. On the contrary, the three next types of problems involve only one area and polygons will represent different suitability measures of the area regarding to different objectives or uses.

On the other hand and identically to punctual or linear actions-based decision problem families, this one can be subdivided into four sub-families:

1. Polygonal actions-based spatio-static decision problems
2. Polygonal actions-based spatio-temporal decision problems
3. Polygonal actions-based spatio-real-time decision problems
4. Polygonal actions-based spatio-sequential decision problems

The first sub-family contains regional physical planning and land-use related problems, where the dynamic nature of the real-world is not considered. Regional physical planning and land-use related problems are usually characterized by impacts and consequences which are dispersed over space and in time. A static formulation of these problems may be suitable to take into account the spatial dispersion of impacts and consequences. However, it is insufficient for handling their temporal evolution.

As for the two previous paragraphs, some physical planning which are characterized with high equity and/or involve parameters of high uncertainty can be approached sequentially in time.

Polygons are also suitable for problems related to the control of moving phenomena (e.g. fire fronts, diseases dispersion, aquatic pollution, etc.). In such problems, a series of polygons will represent the temporal evolution of geographical position, geometry and spatial pattern of the phenomena under focus. Decisions will concern, for instance, the selection of the front of fire to handel first, the region which is more affected with the diseases and which will be treated immediately, etc.

Polygons may also be useful in several environmental applications such as the management of hydraulic dynamic systems. However, in such applications polygons intervene as decision spaces rather than decision actions.

Aside from the general statistical and buffering operations, PointInPolygon and PolygonInPolygon procedures are two typical examples of spatial interaction analysis applied to polygonal actions. PointInPolygon procedure tests if a point is inside a polygon, while PolygonInPolygon procedure tests if a polygon is inside another polygon. Other polygonal typical operations include adjacency tests and surface measurements.

6.2.4 Network actions-based spatial decision problems

Network structures intervene in a variety of real-world problems including shortest path, minimal spanning tree, maximal flow, travelling salesman, airline scheduling, telecommunication, transportation and commodity flow problems. The specificity of network actions in comparison with punctual or linear ones is that they have an inherent spatial information about connectivity which is relevant essentially in road and transportation or drainage network analysis. Identically to previous families, network actions-based one may be subdivided into:

1. Network actions-based spatio-static decision problems
2. Network actions-based spatio-temporal decision problems
3. Network actions-based spatio-real-time decision problems
4. Network actions-based spatio-sequential decision problems

In network related problems, we may be interested in the implementation of a network infrastructure or to its exploitation. Implementation of a network infrastructure has usually long-term impacts and should normally be approached through a temporal formulation. The implementation of a transportation network, for instance, should take into account urban growth, road expansions and soil type, in order to satisfy increasing demands, to avoid congestion and soil slippage or erosion problems. The same remark holds for several other public management and planning problems such as electricity and heat distribution, liquid waste evacuation, etc. Static formulation may be justified in small scale planning problems. In practice, however, networks management problems are usually of high equity and often approached according to a sequential formulation.

Network actions intervene also in a variety of socio-economic applications such as public transportation, automatic route finding in car and truck navigation, commodity flow problems, etc. In such problems, usually several networks are compared mainly in terms of travel time. The specificity of this type of problems is that there is not an explicit selection of a network action but only one action is dynamically constructed. Hence, a dynamic formulation will apply better.

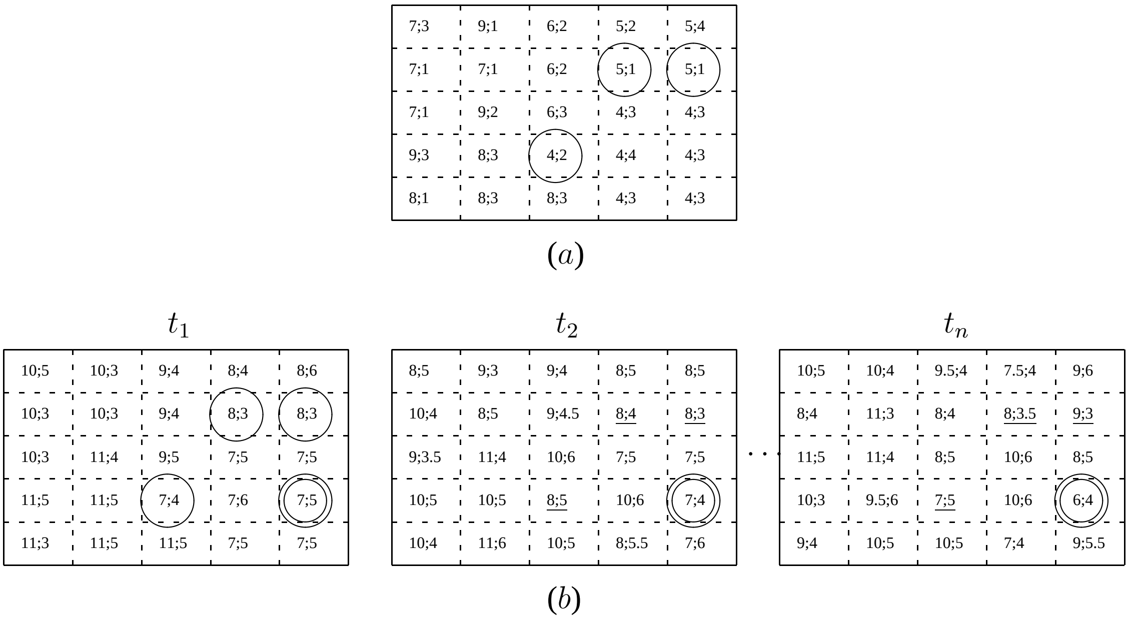
Operations on networks include (Leung, 1997): the set-theoretic operations (e.g. intersection, union, negation, inclusion), spatial and temporal distances, topological operations (e.g. connectivity, accessibility), geometric operations (e.g. length, width, shape, density), pattern, spatial interaction, and functional operations (e.g. shortest path, maximal flow). Inter-entity distances over the network or other measures of connectivity such as travel time, attractiveness, etc., can be used to determine indices of interaction in network-base applications. These operations are much used for determining the location of emergency services or for optimizing delivery routes.

6.3 Some illustrative examples

This section is devoted to present the characteristics of eight sub-families of problems. Firstly, it is important to notice that several didactic and hypothetical examples are discussed in the following paragraphs and the data used in all of them are artificial. We notify also that for the sake of clarity, in all these examples the decision space is schematized through a regular grided form, where punctual, linear, polygonal and network actions are represented respectively with one pixel, a set of linear pixels, a collection of adjacent non-linear pixels, and a combination of individual pixels with several pixel-based linear entities.

Example 1: School location Figure 1.(a) represents a school location problem. In vector-based GIS-like tools a feasible location can completely be defined in terms of its XY-coordinates. In raster-based GIS-like tools, these coordinates are implicitly defined via the position of the cell in the grid. So, for the school location problem, each cell will represent a potential punctual action for locating the school. To each cell we associate two punctual values relative to two sitting criteria, namely implementation cost and average distance from major population centers (values in cells of Figure 1.(a) represent respectively implementation costs and average distances). Here, the two factors are considered to be stable over time, which is convenient to all non strategic location problems in which consequences and impacts of decisions can perfectly be approached via static data. The problem comes down now to the identification of the cell(s) that optimize a certain function f. If we use a weighed sum (regards to its limits) as a decision rule and we consider that the two factors are of equal importance, we obtain three cells that minimize the average sum (circled cells in Figure 1.(a)). This problem involves punctual actions and requires only static information. Thus, it belongs to the family of punctual actions-based spatio-static problems.

Example 2: Nuclear central location Figure 1.(b) represents a hypothetical nuclear waste deposit location problem. Which makes the difference between the situation in Figure 1.(a) and the situation in Figure 1.(b) is that in the latter one, data concerning implementation costs (we suppose that costs in t_2 through t_n are relative to operating and maintenance of the facility to be located because the implementation cost is considered to be committed in t_1) and impacts on environment are considered as time-varying ones and they are expressed as series of values representing measurements of the two factors in different dates. Here, we recognize that siting a nuclear waste deposit is a long-term investment decision implying several impacts on environment that vary across space and in time. For instance, the slope as well as the type of soil will have major roles in reducing or increasing impacts on environment. Taking into account these elements will reduces substantially long-term impacts on environment. As an example, the double circled cell in Figure 1.(b) has better long-term performances than the three circled cells, which may be selected if only data available at t_1 are used. As the previous one, this example involves punctual actions but it requires time-varying data. Thus, it belongs to the family of punctual actions-based spatio-temporal problems.



7;3	9;1	6;2	5;2	5;4
7;1	7;1	6;2	5;1	5;1
7;1	9;2	6;3	4;3	4;3
9;3	8;3	4;2	4;4	4;3
8;1	8;3	8;3	4;3	4;3

(a)

t_1

10;5	10;3	9;4	8;4	8;6
10;3	10;3	9;4	8;3	8;3
10;3	11;4	9;5	7;5	7;5
11;5	11;5	7;4	7;6	7;5
11;3	11;5	11;5	7;5	7;5

t_2

8;5	9;3	9;4	8;5	8;5
10;4	8;5	9;4.5	8;4	8;3
9;3.5	11;4	10;6	7;5	7;5
10;5	10;5	8;5	10;6	7;4
10;4	11;6	10;5	8;5.5	7;6

. . .

t_n

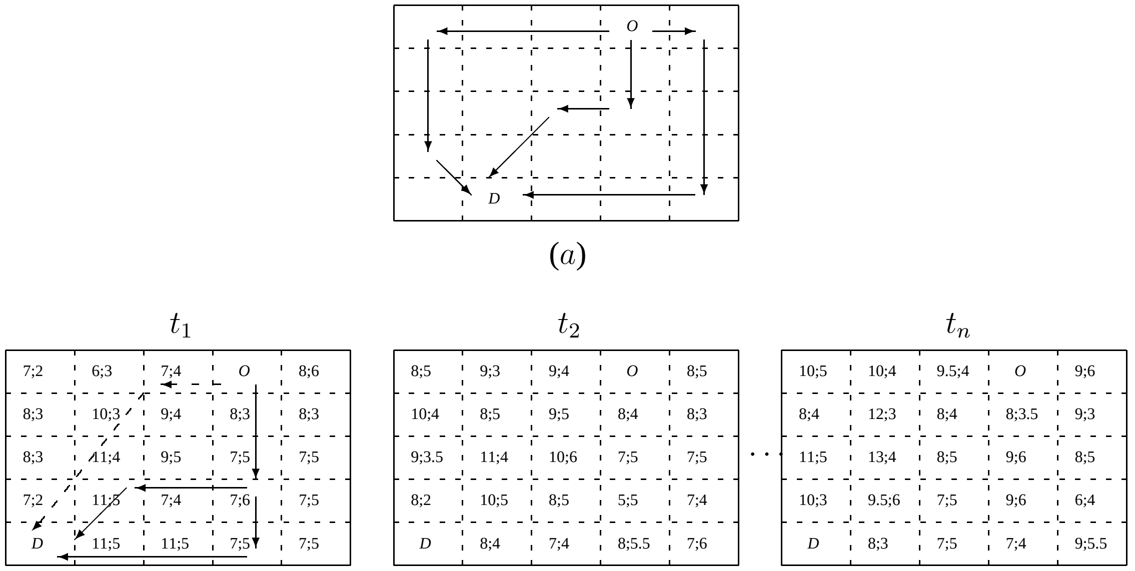
10;5	10;4	9.5;4	7.5;4	9;6
8;4	11;3	8;4	8;3.5	9;3
11;5	11;4	8;5	10;6	8;5
10;3	9.5;6	7;5	10;6	6;4
9;4	10;5	10;5	7;4	9;5.5

(b)

Figure 1: Schematic representation of school (a) and nuclear central (b) location problems

Example 3: Route selection In Figure 2.(a) we present a simple trip itinerary selection problem. The objective is to select the route to follow during the trip. In most of individual spatial decision-making problems, decisions are usually tackled within ad hoc manner on the basis of simple heuristics and/or past experiences. In the route selection problem, several routes are available (in Figure 2.(a) there are three routes; each one is represented as a collection of linear cells) and the adoption of a particular route will depend on punctual information issued mainly from past experiences. In this example, such information are sufficient to represent the decision environment because there is no need to take into account future evolution of real-world. This may be explained with the immediate nature of the consequences of decisions and the low cost of a 'poor' decision. In this example, the cost of a 'poor' decision is simply to put more time to arrive. This problem concerns the selection among several linear actions and evokes non strategic decision having immediate consequences. Accordingly, it belongs to linear actions-based spatio-static decision problems family.

Example 4: Highway construction Considering the problem of the selection of a corridor for implementing a new highway illustrated in Figure 2.(b). The objective is to select the corridor that applies for future demands and respects the environment. The two values in cells of the matrix of Figure 2.(b) represent respectively implementation/operating and maintenance costs and impacts on environment levels. Each two values are relative to the part of the corridor which is inside the cell. If we consider only the data available at t_1, one may have the three corridors represented in the figure. If we suppose also that the decider, basing on the data available at t_1, selects the corridor that is represented with dashed arrows, it may happen that in future time, this corridor supposed currently as the 'optimal' one, generates more negative impacts than the two other ones. We can not avoid a such eventuality only if we dispose of solid predictions of the sates of real-world in the future that are established by well-experienced specialists. In our example, we dispose of three matrix, the first is relative to current time, the two others are predictions of real-world in two future dates. As in the nuclear central location problem, the question that raises here is to use these predictions to select the corridor that has better long-term performances. This illustrative example requires the use of dispersed data to select line-based actions. Consequently, it belongs to linear actions-based spatio-temporal problems family.



(a)

t_1

7;2	6;3	7;4	*O*	8;6
8;3	10;3	9;4	8;3	8;3
8;3	11;4	9;5	7;5	7;5
7;2	11;5	7;4	7;6	7;5
D	11;5	11;5	7;5	7;5

t_2

8;5	9;3	9;4	*O*	8;5
10;4	8;5	9;5	8;4	8;3
9;3.5	11;4	10;6	7;5	7;5
8;2	10;5	8;5	5;5	7;4
D	8;4	7;4	8;5.5	7;6

. . .

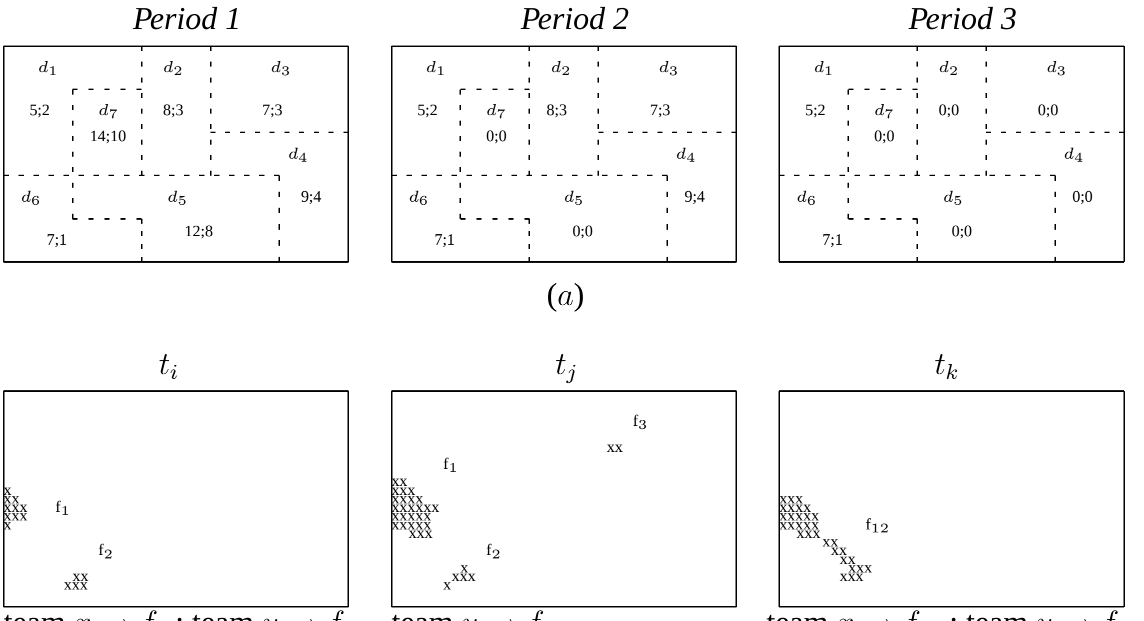
t_n

10;5	10;4	9.5;4	*O*	9;6
8;4	12;3	8;4	8;3.5	9;3
11;5	13;4	8;5	9;6	8;5
10;3	9.5;6	7;5	9;6	6;4
D	8;3	7;5	7;4	9;5.5

(b)

Figure 2: Schematic representation of route selection (a) and highway construction (b) problems

Example 5: Historic zone restoration Figure 3.(a) represents the different districts of an historic zone where an ambitious project of restoration is envisaged. The project has considerable financial requirements and may have several undesirable impacts on the socio-economic activities (e.g. problem of congestion), mainly if interventions are dispersed around all the region. To avoid such situations, local authorities have defined an action plan of fifteen years divided into three periods, each of five years. The problem now is to classify the 7 districts into three groups in priority order. We suppose that the two criteria considered are: the total number of sites to be restored and the number of highly priorate sites (measurements of the two criteria are depicted in Figure 3.(a)). Here, polygons represent potential actions which should be regrouped into three ordered classes. In the first five years, the priory districts (d_5 and d_7) are restored. Designing which are the next districts to restored may be defined in the beginning of the first period or in future time. In the first case, we may select districts d_2, d_3 and d_4 to be restored in period 2 and districts d_1 and d_6 to be restored in period 3 (notice that the number of priority zones may increases over time). In this problem, actions are schematized through polygonal structures. In addition, a series of decisions is required to select the different districts to handel in each period. Thus, it belongs to the polygonal actions-based spatio-sequential problems family.



(a)

team $x \rightarrow f_1$; team $y \rightarrow f_2$ team $y \rightarrow f_3$ team $x \rightarrow f_{12}$; team $y \rightarrow f_{12}$

(b)

Figure 3: Schematic representation of historic zone restoration (a) and fire fighting (b) problems

Example 6: Forest fire fighting In fire fighting problems we are subject to several constraints related to time pressure, unpredictability of changes, and non availability of sufficient resources. Figure 3.(b) represents evolution of the decision process in a hypothetical fire fighting problem where only two fire fighting teams (x and y) are available. This figure schematizes the evolution of fire fronts over space and in time. The decider is called to coordinate the two teams in order to extinct fires as soon as possible and to minimize damages. He continuously receives information concerning the evolution of fronts from different controlling centers. In this problem, fronts take the form of polygons which their form and position change across space and time, and where decisions take the form of "affect team x to front f". At time t_i, two fire fronts are observed (f_1 and f_2) and the decision was to affect team x to front f_1 and team y to front f_2. Then at time t_j the decider sent team y to a new fire front, f_3. As time progress, the timing of decisions will be more relevant for the achievement of objectives. The situation is complicated with the fact that decision taken in current time will constrain future possible decisions. In our example, affecting team y to front f_3 had permitted to eliminate this front but a such decision will complicate the situation around front f_2. In fact, at time t_k, front f_2 is extended to constitutes, with front f_1, front f_{12}. The process will continues until the extinction of all fire fronts. This problem involves real-time decisions concerning the control of moving phenomena that take the form of polygon structures. Consequently it belongs to the polygonal-actions spatio-real-time problems family.

Example 7: Track navigation In track navigation problems, the transportation network is dynamically changing (in terms of traffic rate, climatic conditions, accidents, etc.). In such problems deciders are often subject to time pressure constraint. The specify of this type of problems is that there is not an explicit selection of a network. Instead, only one network is dynamically constructed over time. In reality, the decisions in this problem concern the selection of the route to add to the network being constructed and each of these decisions will simply create an 'instance' of the required global network. However, the ultimate objective is not select individual routes but to reach different demand points as soon as possible. In this problem attention is focalized on the dynamic definition of a transportation network and consequently it belongs to the family of network actions-based spatio-real-time decision problems.

Example 8: Subway implementation The construction of a subway is of a high equity. As for most strategic pubic planning problems, this one is approached sequentially in time. Accordingly, we should define an action plan of several periods. In each period, the decision concerns the implementation of a sub-network. The problem will comes down to the selection of the sub-network to implement at each period of time. This will depend on several technical, economic and social factors. For instance, sub-networks may be compared in terms of their implementation costs, their role in reducing negative effects of current congestion problems, and in terms of their coverage levels. In this example, the initial problem is decomposed into a series of sub-problems in which actions take the form of network structures. Consequently, it belongs to the family of network actions-based spatio-sequential decision problems.

The characteristics of the above-cited typical problems are summed-up in Table 4.

Problem	*Characteristics*	*Family*
School location	Punctual potential actions Short or medium-term consequences	Punctual actions-based spatio-static problem
Nuclear central location	Punctual potential actions Long-term consequences	Punctual actions-based spatio-temporal problem
Route selection	Linear potential actions Immediate consequences	Linear actions-based spatio-static problem
Highway construction	Linear potential actions Long-term consequences	Linear actions-based spatio-temporal problem
Historical zone restoration	Polygonal potential actions High equity	Polygonal actions-based spatio-sequential problem
Forest fire fighting	Polygonal potential actions Dynamically changing environment	Polygonal actions-based spatio-real-time problem
Track navigation	Network potential actions Dynamically changing environment	Network actions-based spatio-real-time problem
Subway implementation	Network potential actions High equity	Network actions-based spatio-sequential problem

Table 4: Characteristics of the illustrative examples

7 Problems implying complex potential actions

In many real-world applications, one may be called to represent actions with a combination of two or more atomic entities. In schools partitioning problem, for instance, decision actions can be assimilated to a combination of points and polygons where points represent schools and polygons represent zones to serve. A set of ’point-point’ composed actions may represent potential paths in a shortest path identification problem and a set of composed actions of ’point-network’ type may schematize different feasible locations, each one belongs to a different transportation network. Table 5 provides some other examples of problems where complex actions are required.

One particular composed action is the one based on a map structure. Map structures are relevant mainly in spatial problems that are related to the control of (non real) spatial phenomena. An illustrative example of representing decision spatial actions using map structures is provided in Janssen and Herwijnen (1998). The authors have proposed several transformation and aggregation methods in order to represent the performances of different policies of antipollution fights in the ’Green Heart’ region of the Netherlands. The results of the transformation and aggregation operations have been presented in performance maps, which represent the relative quality of the different policies along with their spatial patterns. These maps are then used as inputs to the evaluation step.

Another example of using map structures to represent decision actions is furnished by Sharifi et *al.* (2002), where the authors have interested to the problem of relocating the boundary between the ’Tunari National Park’ and the ’Cochabamba City’ (in Bolivia) in order to avoid spontaneous illegal settlements in between the park and the city. Four different maps, each represents a possible approach to address the problem and satisfy the objectives of stakeholders, are generated and compared with current situation.

Potential Action	*Typical problem*
Point-Point	Shortest path problem: pairs of points represent different paths
Point-Line	Bus stops location: lines represent routes and points are candidate stations
Point-Polygon	School partitioning problem: points schematize schools while polygons represent zones to serve
Point-Network	Location in a distribution network: points represent different distribution sites (e.g. supermarkets) in the distribution network
Line-Line	Routes intersection: linear objects schematize the different routes
Line-Polygon	Agriculture preservation: rivers are represented with lines and zones to preserve with polygons
Line-Network	Adding a new route in a distribution network: lines are potential arcs to be included in a distribution network
Polygon-Polygon	Hierarchical zoning: administrative zoning where districts, departments, etc., take the form of hierarchical polygons
Polygon-Network	Industrial zone location: polygons schematize different potential zones in a transportation network
Network-Network	Correspondence between networks in public transportation: between an underground and bus networks, for instance
Map	Antipollution fight policy choice: each map represents the spatial pattern of a potential policy
Map-Point	Regional planning problem: maps represent different potential regions for implementing a new regional hospital and points represent potential sites for locating the hospital

Table 5: Examples of complex spatial potential actions (some ones are reproduced from Malczewski (1999))

Map-based actions are also suitable for applications in which a strong spatial relation between elements of the decision space (e.g. spatial contingency and adjacency in redistricting problems, spatial compactness in land use allocation problems, composition relations in administrative partitioning problems) should be verified. An example is the redistricting problem where attention is focalized on the definition of a zoning (representing, for example, administrative, commercial or service zones) that verifies at best the spatial contingency propriety between neighbor zones and, at the same time, eliminates intersections between these zones and avoids holes. In such a problem, maps constitute an excellent support for the presentation of potential partitions and for their visual evaluation and then their comparison.

Complex actions should verify several spatial relations (e.g. proximity, appurtenance, minimum distance separation for new development from existing livestock facilities) among its atomic entities. These relations will serve as inclusion/exclusion criteria by which one atomic action is included or not to another atomic action representing the new decision space.

Generally, the generation of complex actions begins by defining more complex atomic actions (e.g. networks) on which less complex atomic actions are defined (e.g. in the school partitioning problem cited above, we should normally define polygons and then associate to each polygon a punctual location action). In some cases, the order by which actions are defined may be imposed by the nature of the problem (e.g. locating restaurants in pre-existing highways, where highways with high potential demands are selected first followed by punctual locations on these highways).

Composed actions-based spatial decision problems are not explicitly included in the typology detailed in §6. Nevertheless the typology is still adequate to describe most of them. Indeed, these last ones are usually decomposed into a series of basic problems, each one involves only one atomic action type. Considering, for instance, that our initial problem involves composed actions which are constituted of two atomic actions e_1 and e_2 that should verify a spatial relation r. This problem can be decomposed into two steps (see Figure 4):

- in the first step we resolve a first sub-problem involving actions of type e_1, and
- in the second step we resolve another sub-problem involving actions of type e_2.

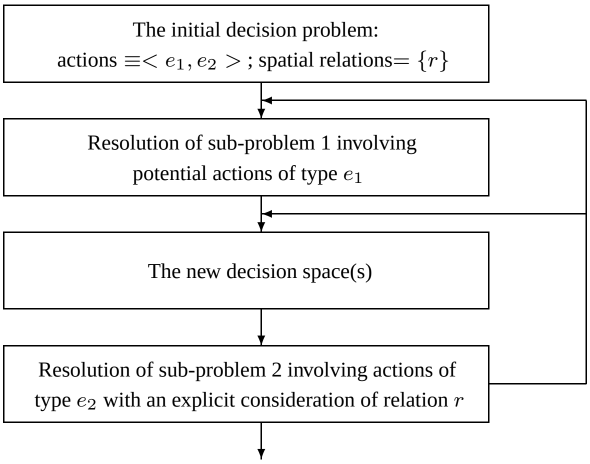

The initial decision problem:
actions $\equiv< e_1, e_2 >$; spatial relations$= \{r\}$
Resolution of sub-problem 1 involving
potential actions of type e_1
The new decision space(s)
Resolution of sub-problem 2 involving actions of
type e_2 with an explicit consideration of relation r

Figure 4: A process for handling complex potential actions-based spatial decision problems

The resolution of the first sub-problem permits to define the new decision space(s) over which the second sub-problem is defined and resolved. The resolution of the second

sub-problem should take into account spatial relation r. We remark that the process illustrated in Figure 4 is an interactive one. This enables the decider and/or analysts (i) to modify the input data and (ii) to repeat the process as many as necessary.
To better illustrate this idea, consider the location-allocation problem illustrated in Figure 5. In this example we look to locate n service points in a given region. The initial problem involves 'point-map' composed actions where points represent the geographical position of the different service points and maps represent the different possible partitions of the study region. This problem may be decomposed into two sub-problems as follows. The first sub-problem is a partitioning one, which aims to subdivide the region into several homogenous zones. It corresponds to a map actions-based problem. The resolution of this problem permits to define the decision spaces of n location sub-problems. These n sub-problems correspond to n punctual actions-based problems. In this example, the spatial relation r may be an appurtenance relation, i.e., each service point belongs necessarily to the zone that it services. It is important to notice that map-based actions are normally of complex nature but in this example they may be considered as atomic ones because each map represents a unique manner for subdividing the study region into homogenous zones.

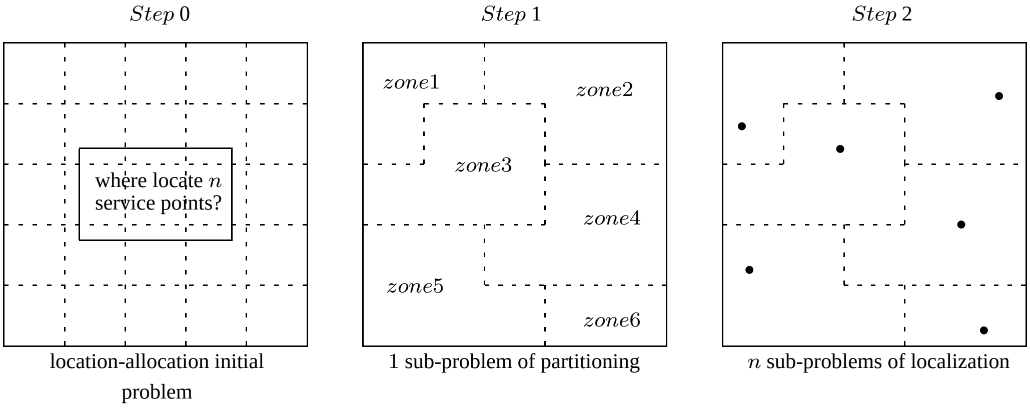

$Step$ 0
$Step$ 1
$Step$ 2
where locate n service points?
$zone1$
$zone2$
$zone3$
$zone4$
$zone5$
$zone6$
location-allocation initial problem
1 sub-problem of partitioning
n sub-problems of localization

Figure 5: Location-allocation illustrative example

Finally, we notice that modes for approaching spatial decision problems detailed in §4 as well as the descriptions of spatial decision problems provided in §6.1 apply also to problems implying complex spatial actions. We particularly notice that data structures defined in §6.1 are suitable to this type of problems. They, however, require more complex modelling and resolution techniques. Equally, evaluation and comparison of complex actions require more complex operators and analysis routines. For instance, techniques as overlay[4], spatial filtering and zoning are often applied to actions based on map structures.

[4] It refers to arithmetic and logic operations that are applied to maps representing different geographic themes. Overlapping is likely the most used operation in GIS-like tools.

8 Concluding remarks

This paper is a tentative for classifying spatial decision problems. Concretely, we have presented a typology obtained through a crossover of actions' types with different possible modes for approaching spatial decision problems. The proposed typology covers most of spatial decision problems that involve atomic actions. It permits also to represent and model several spatial problems that involve complex actions. The idea is to decompose the initial problem into a series of basic spatial problems that involve atomic actions. Notice, however, that it is by no means to suppose that the typology encompasses all spatial decision problems.

The proposed typology overcomes several limits of typologies of section 2. Indeed, it has at least the following merits. First, it is useful for understanding the specificities of spatial decision problems. Second, the typology facilitates the choice of spatial and temporal operators and analysis routines required to each problem type. In fact, these operators and routines are essentially determined by the type of spatial objects by which real-world spatial entities (and consequently spatial potential actions) are conceived and digitally represented. Third, it provides tools for formulating spatial problems and representing their dynamics by explicitly integrating the temporal dimension, which is necessary to convenably characterize these problems. Finally, it constitutes a suitable framework for multidisciplinary researches because it is based on simple concepts and paradigms which are devoid of any socio-economic or environmental contexts, and on which researches from different disciplines agree.

Furthermore, the typology is particularly useful to develop multicriteria evaluation-based spatial decision-aid tools. In fact, the most used practice in multicriteria spatial decision-aid consists in representing decision actions in terms of spatial entities. Practically, the typology will provide convenable framework (a) for generating, evaluating and comparing potential actions, and (b) for dealing with conceptual, methodological and technical questions related to the undertaking of the dynamics of basic elements of multicriteria evaluation models (i.e. actions, criteria and preferences) in many real-world applications.

Equally, the typology is useful to develop spatial decision support systems (or SDSS). Specifically, it helps to construct a general framework for classifying analysis, modelling and resolution techniques according to their suitability to different problem formulations. This framework will represent the first step towards the development of a model base management system to be incorporated into the SDSS. Its role is to assist analysts and deciders to select the adequate technique(s) for the problem under focus. In fact, a large variety of structured models including statistical methods, mathematical models, heuristic procedures, algorithms, and so on, are available in GIS-like tools, and usually analysts and deciders come across difficulty in selecting the relevant model to use. A well-established framework can therefore be used for a systematic organization of analysis and modelling tools through, for instance, IF-THEN rules or YES-NO questions (Leung, 1997), inside the SDSS.

Several of the topics that are mentioned in this paper require further attention. In particular, attention should be addressed towards the elaboration of an exhaustive description of spatial operators and analysis routines which are suitable to each type of actions as well as towards the identification of modelling and resolution techniques and tools that are convenable to each family of problems. Additionally, the process of decomposition of problems involving complex actions discussed in §7 requires a more elaborated study. All these elements will be dealt with in our future research where we intend to focalize on a particular family of problems, namely linear-based spatio-temporal decision problems. Our objective is to conceive and develop a multicriteria spatial decision-aid tool devoted to linear infrastructure management and planning problems.

Acknowledgement

The authors are really thankful to *Professor* Bernard Roy for his valuable comments on an earlier version of this paper.

References

[1] Y. Bedard. Géomatique et systèmes d'information à référence spatiale en milieu municipal. *Journées de l'Association de Géomatique Municipale du Québec (AGMQ)*, Laval, Mars 1991.

[2] C. Claramunt, M. Thériault, and C. Parent C. A qualitative representation of evolving spatial enties in two-dimensional spaces. In *GIS Research UK: $5th$ National Conference*, Leeds, $9th$-$11th$ April 1997.

[3] A. Diederich. Sequential decision making. In T. Marley, editor, *International Encyclopedia of the Social and Behavioral Science: Methodology*, Mathematics and Computer Science. Pergamon, Amsterdam, 1999.

[4] A. Frank. Socio-economic units: Their life and motion. In A. Frank, J. Raper, and J.-P Cheylan, editors, *Formalising and reresenting spatial-temporal change in spatial socio-economic units*, GISDATA. Taylor & Francis, London, 1999.

[5] P. Jankowski. Spatial decision making—an overview, Spatial Decision Support System Course, Idaho University, USA, 2003. [Available at http://geolibrary.uidaho.edu/courses/Geog427/Lectures/1/].

[6] R. Janssen and M. van Herwijnen. Map transormation and aggregation methods for spatial decision support. In E. Beinat and P. Nijkamp, editors, *Multicriteria*

Analysis for Land-Use Management, Number 9 in Environment and Management Series, pages 253–270. Kluwer Academic Publishers, Dordrecht, 1998.

[7] C.P. Keller. Decision support multiple criteria methods. In *NCGIA Core Curriculum, National Center for Geographic Information and Analysis*, University of California, Santa Barbara, 1990.

[8] S. Lardon, T. Libourel, and J.-P. Cheylan. Concevoir la dynamique des entités spatio-temporelles. *Revue Internationale de Géomatique*, 1(9):45–65, 1999.

[9] Y. Leung. *Spatial analysis and planning under imprecision*. North-Holland, Amsterdam, 1988.

[10] Y. Leung. *Intelligent spatial decision support systems*. Advances in Spatial Science. Springer-Verlag, Berlin, 1997.

[11] D.J. Maguire, M.F. Goodchild, and D.W. Rhind, editors. *Geographic information systems: Principles and applications*. Longman Scientific and Technical, Harlow, 1991.

[12] J. Malczewski. *GIS and mutlicriteria decision analysis*. John Wiley & Sons, New York, 1999.

[13] G. Munda. *Multicriteria evaluation in a fuzzy environment: theory and applications in ecological economics*. Physica-Verlag, Heidelberg, Germany, 1995.

[14] M.A. Sharifi, W. van den Toorn, A. Rico, and M. Emmanuel. Application of GIS and multicriteria evaluation in locating sustainable boundary between the Tunari National Park and Cochabamba city (Bolovia). *Journal of Multi-Criteria Decision Analysis*, 11(3):151–164, 2002.

[15] H.A. Simon. *The new science of management decisions*. Random House, New York and Evanston, 1960.

[16] E. Stefanakis and T. Sellis. Towards the design of a DBMS repository for the application domain of GIS. In *Proceedings of the 18th International Cartographic Conference*, pages 2030–2037, Stockholm, Sweden, 1997.

An Overview of What We Can and Cannot Do with Local Search

Petros Christopoulos[*], Vassilis Zissimopoulos[*]

Résumé

Etant donné qu'on ne connaît pas un algorithme efficace pour résoudre les problèmes d'optimisation NP-difficiles, on développe plusieurs algorithmes approchés. Parmi ces algorithmes est la Recherche Locale qui est une méthode générale. On considère une structure de voisinage de solutions d'un problème d'optimisation et au lieu de trouver la meilleure solution dans le domaine, nous trouvons une solution, appelée optimum local, qui est la meilleure dans ce voisinage. Ainsi, l'heuristique standard de la recherche locale commence par une solution initiale et se déplace à une meilleure solution voisine pour aboutir à un optimum local. Cette simple méthode se démontre en pratique très performante produisant des solutions de bonne qualité dans de temps d'exécution raisonnable.

Le but principal de ce travail est de faire une synthèse du travail théorique qui est réalisé sur les limites de la recherche locale en général et son efficacité d'approximation pour de problèmes spécifiques. Ainsi, d'un côté nous présentons la théorie de PLS-complétude et nous montrons que pour un problème PLS-complet l'heuristique de la recherche locale standard nécessite dans le pire de cas de temps exponentiel. Nous montrons aussi que s'il est NP-difficile d'obtenir une $\varepsilon-$approximation d'un problème d'optimisation alors il n'existe pas de voisinage qui conduit à un optimum local $\varepsilon-$proche d'un optimum global, sauf si NP=co-NP. De l'autre côté, nous présentons plusieurs exemples de problèmes NP-difficiles pour lesquels certains voisinages peuvent garantir des optima locaux $\varepsilon-$proche d'un optimum global. Cette garantie est souvent, la meilleure qu'on puisse obtenir pour certains problèmes par n'importe quel algorithme. L'algorithme de la recherche locale est pseudopolynomial. Par conséquent, lorsque les problèmes sont sans poids ou avec des poids polynomialement bornés l'algorithme atteint un optimum local en temps polynomial.

Mots-clefs : recherche locale, PLS-complétude, approximation

[*] Department of Informatics and Telecommunications, University of Athens, 15784 Athens, Greece. `{p.christopoulos, vassilis}@di.uoa.gr`

Résumé

Since we do not know any algorithm to efficiently solve the NP-hard optimization problems, a lot of approximation algorithms have been evolved. A general method for this purpose is Local Search. One assumes a neighboring structure between the solutions of an optimization problem and wants to find a solution that is the best in its neighborhood, called a local optimum, instead of the best solution in the domain. So, the standard local search heuristic starts from an initial solution and keeps moving to some better neighbor until it reaches a local optimum. This simple method turns out to be very successful in practice both in its running time performance and on the quality of the solutions that produces.

The main purpose of this work is to sum up the theoretical work that has been done concerning the limits of local search in general and its proven approximation efficiency for particular problems. Hence, on the one hand we present the PLS-completeness theory and show that for the PLS-complete problems the standard local search heuristic takes exponential time in the worst case. We also show that if it is NP-hard to ε-approximate an optimization problem then there is no neighborhood which produces local optima only ε-close to global optima, unless NP=co-NP. On the other hand, we present numerous of examples of NP-hard optimization problems that under appropriate neighborhoods guarantee local optima ε-close to the global optima. Such guarantees are, in many cases, between the best ones for these problems by any algorithm. Local search heuristic is pseudopolynomial, so when the problems are unweighted or with polynomially bounded weights it reaches a local optimum in polynomial time.

Key words : local search, PLS-completeness, approximability

1 Introduction

Local Search is a widely used general method to approximately solve NP-hard optimization problems. Roughly speaking, an optimization problem has a solution set and a cost function, which assigns a value to each solution. Our aim is to find an optimal solution, that is a solution with the smallest or greatest value. We can obtain a *local search heuristic* for an optimization problem by superimposing a neighborhood structure on the solutions, i.e. we define for each solution a set of neighboring solutions. The heuristic starts from an initial solution, which is constructed by another algorithm or is chosen randomly, and keeps moving to some better neighbor as long as there is one. So the heuristic stops when it reaches a *locally optimal solution*, i.e. one that does not have any better neighbor.

An important theory for local search has been developed, about both its complexity and its approximation ability. This theory will be our issue for the first part of this paper

(Sections 4 to 6) and more particularly the answers that have been given to the following questions:

- What is the complexity of the local search heuristic? (that is, how fast can we find a local optimum?)
- What is the quality of the solutions that a local search heuristic can give? (that is, how close to the global optimum can be the local optimum).

Generally, for many interesting problems, the complexity of finding a locally optimal solution remains open, that is we do not know whether it can be found in polynomial time or not, by any algorithm. The complexity class PLS (Polynomial-time Local Search), which is presented in Section 4, was defined by Johnson, Papadimitriou and Yannakakis, in [1], exactly for the purpose of grouping such problems, provided that their neighborhood is polynomially searchable. This is the least requisition which is satisfied by the neighborhoods used in the common local search heuristics. PLS class lies somewhere between P and NP. It has been shown that a lot of important local search problems, which we will describe in Section 3, are complete for PLS under a properly defined reduction, that is why PLS characterizes the complexity of the local search problems in the same sense that NP characterizes the complexity of the "hard" optimization problems. Furthermore, we can analyze the complexity of many popular local search heuristics with the aid of the PLS-completeness theory.

Hence, through this theory, in Section 5, a (negative) answer to the first question, that of complexity, is given for a lot of important local search problems (for example TSP/k-Opt, GP/Swap and Kernighan-Lin, Max-Cut/Flip, Max-2Sat/Flip, Stable configurations for neural networks). However, there are still "open" problems, such as TSP/2-Opt, for which we do not know whether they can be solved polynomially by the local search heuristic.

About quality, we will see in Section 6 a theorem restrictive for the approximability of the (globally) optimal solutions of NP-hard problems, with the local search method. Additionally, we will see that it is also impossible, given a PLS problem to guarantee to find solutions ε-close to a local optimum, in polynomial time, by the standard local search heuristic. However, we can find solutions that are $(1+\varepsilon)$ times smaller (resp. bigger), for minimization (resp. maximization) problems, from their neighbors by an FPTAS.

The planning of a good neighborhood and hence of a successful local search heuristic is mainly achieved by experimental technics, trying to keep a balance between the quality of the solution and the time needed to be found. In Section 7, we describe many NP-hard problems for which the local search heuristic provably gives solutions ε-close to their global optima, in polynomial time, using appropriate neighborhoods. We also, present some characteristic proofs of this kind.

The importance of local search is based on its simplicity and its surprising efficiency in "real-life" problems. Despite its main use with NP-hard problems, local search is also used in many other cases that we will refer in Section 8. Hence, there are polynomial problems for which local search heuristics are better, in practice, than other polynomial algorithms, such as Linear Programming and Maximum Matching. For these problems we consider neighborhoods which guarantee that their local optima coincide with the global optima. Another use of local search is as a tool in proofs of existence of an object. For example, see in subsection 3.6 the proof that there is always a stable configuration in neural networks of the Hopfield model. The theory of local search was recently applied in Game Theory giving some interesting first results. Finally, local search has a powerful extension, non-oblivious Local Search, and is also the base of the most metaheuristics such as Tabu Search and Simulated Annealing.

The following section sets up a formal framework for local search and provides some useful definitions and questions, for the rest of this work.

2 Framework

A *general computational problem* Π has a set D_Π of instances and for each instance $x \in D_\Pi$ there is a set $\mathcal{J}_\Pi(x)$ of corresponding possible feasible answers. An algorithm solves problem Π if on input $x \in D_\Pi$ it outputs a member y of $\mathcal{J}_\Pi(x)$ or in case that $\mathcal{J}_\Pi(x)$ is empty, it reports that there is no such y. Inputs (instances) and outputs are represented as strings upon a finite alphabet, which without loss of generality can be considered to be $\{0, 1\}$.

There are different types of computational problems depending on the size of their output sets $\mathcal{J}_\Pi(x)$. The most general kind is a *Search* problem where each $\mathcal{J}_\Pi(x)$ can be consisted of zero, one or more elements. If $\mathcal{J}_\Pi(x)$ is non-empty for $x \in D_\Pi$ the problem is called *Total*. If $|\mathcal{J}_\Pi(x)| \leq 1$ for every x, then it is called *Functional*. An important special case of *Total Functional* problems, i.e. those for which holds $|\mathcal{J}_\Pi(x)| = 1$ is the set of the *Decision* problems, where for every instance the unique answer can be either "Yes" (1) or "No" (0). For instance, does a given graph has a Hamiltonian cycle?

An instance x of an *Optimization* problem Π is a pair $(\mathcal{J}_\Pi(x), f_\Pi(i, x))$, where $\mathcal{J}_\Pi(x)$ is any feasible solution set and $f_\Pi(i, x)$ is a *cost function* for every $i \in \mathcal{J}_\Pi(x)$, i.e. a mapping $f : \mathcal{J} \rightarrow \mathbb{R}$. The problem is to find an $i^* \in \mathcal{J}$ such that $f(i^*, x) \leq f(y, x)$ for a minimization problem or $f(i^*, x) \geq f(y, x)$ for a maximization problem, for every $y \in \mathcal{J}$. Such a point i^* is called a *global optimal solution* of the specific instance. The cost function assigns a cost on every solution of the instance of the problem, which is usually a positive integer. An optimization problem is a set $\mathcal{I}$ of such instances. These problems are in fact a kind of search problems, since they turn up by applying a cost function on the latter (there can, also, be more than one optimal solutions with, the same, optimal cost).

Optimization problems, now, are divided in two categories: those whose solutions are consisted of variables, which take discernible values and are called *Combinatorial* (for example Integer Linear Programming, Graph Partitioning) and those whose solutions are consisted of variables, which take continuous values and are called *Continuous* (for example, Linear Programming[1]).

Defining a neighborhood structure on the set of the solutions of a Combinatorial Optimization problem, we obtain a *Local Search* problem. The Local Search problem is this: Given an instance x, find a *locally optimal solution* $\hat{ß}$. That is, a solution which does not have any strictly better neighbor (smaller cost in minimization problems or greater cost in maximization problems). This is a Search problem, too, because an instance may have many locally optimal solutions.

Given an instance x, a set $\mathcal{N}_\Pi(i, x)$ of neighboring solutions is assigned on every solution $i \in \mathcal{J}_\Pi(x)$. The neighbors are determined each time from the current instance x and the current solution i. From a Combinatorial Optimization problem and different neighborhood functions arise different Local Search problems, for this reason we symbolize them as $OP/\mathcal{N}$.

Even though the term *neighborhood* implies that the neighboring solutions are "close" to each other in some sense, i.e. that we can obtain one solution from the other by a small perturbation, this need not generally be the case. The neighborhood function can be very complicated and it does not even need to be symmetric. It is possible for a solution i to have a neighboring solution j, when j does not have i for neighbor. For example, consider the *Graph Partitioning* problem. A very simple neighborhood is *Swap*, which is defined as follows: a partition (A, B) of the set of the nodes V into two equal parts has as neighbors all the partitions that are obtained by swapping one node from A with one node from B. A much more complex neighborhood is being searched by the *Kernighan-Lin* algorithm in [2]. This neighborhood is not symmetric and depends on the weights of the edges. Additionally, if a partition is locally optimal under the Kernighan-Lin neighborhood then it will be locally optimal under Swap, too. Hence, finding local optima under Kernighan-Lin is at least as difficult as it is under Swap. Surprisingly, as we will see later, the two problems are actually polynomially equivalent.

In general, the more powerful a neighborhood is the harder one can search it and find local optima, but probably these optima will have better quality. The most powerful neighborhood is the *exact* neighborhood. In an exact neighborhood local and global optima are the same, and hence the local search problem coincides with the optimization problem. Of course, in every optimization problem one can make local and global optima

[1] Linear Programming, in particular, can also be considered as a Combinatorial Optimization problem, since the solution that we are looking for belongs in the finite set of the vertices of a polytope.

to coincide by selecting sufficiently large neighborhoods. But the problem is that in such a case it will be difficult for one to search the neighborhood. That is, finding whether a solution is locally optimal and if not finding a better neighbor will take exponential time. Therefore, an essential consideration is that we must be able to search the neighborhood efficiently.

For the problems that we will discuss later the following framework is used: A General Local Search problem Π consists of a set of instances D_Π and for each instance $x \in D_\Pi$ there is a solution set $\mathcal{J}_\Pi(x)$, a cost function $f_\Pi(i, x), i \in \mathcal{J}_\Pi(x)$ and a neighborhood function $\mathcal{N}_\Pi(i, x), i \in \mathcal{J}_\Pi(x)$. The problem is as follows: Given an instance x, find a locally optimal solution. *Note that the use of any algorithm is allowed, not necessarily of a local search algorithm.*

Besides the local search problem itself we will also discuss about the corresponding heuristic. The *standard local search algorithm* does the following actions: starts from an initial solution (which is obtained either randomly or by another algorithm), moves to a better neighboring solution and repeats the previous step, until it reaches a solution $\hat{ß}$, which has no better neighbors and is called *local optimum*. Notice that the complexity of a local search problem can be different from that of its corresponding standard local search heuristic. Consider, for instance, Linear Programming which is a problem in P, but Simplex, the local search algorithm used to solve it, has a worst-time complexity exponential. The two issues that we are concerned about a local search heuristic is its complexity (Section 5) and the quality of the solution that it finds, i.e. how close to the global optimum are the local optima (Section 7).

Assuming that the complexity of one iteration is polynomial, in Section 5 we are looking for the complexity in the worst case (as function of the size of the instance, for the instances and all their initial solutions) of the standard local search algorithm with a specific pivoting rule. This problem is called *Running Time Problem*. The running time of the local search algorithm, whether it is polynomially bounded or not, depends mainly on the number of the iterations.

The *pivoting rule* is the rule that is used by the heuristic to choose to which specific better solution, among all the better neighboring solutions that a solution might have, to move. In the examination of the local search algorithms we want to analyze the complexity for different pivoting rules and to find the best one.

Notice, also, that the standard local search algorithm is pseudopolynomial. This means that its complexity in the worst case depends from the numbers (weights, costs) that exist in the instance. If the number of the different solution costs is polynomial, then there will be, at most, a polynomial number of iterations and the algorithm will converge in polynomial time (this holds since, as we said, in each step we always go to a better neighbor). For example consider the non-weighted versions of many optimization problems, such as Graph Partitioning without weights on the edges of the graph. In the general case, where an exponential range of solution costs exists, there is no a priori bound better than the exponential. In such a case we want to know if the local search problem can be solved polynomially or not.

Finally, note that the quality of a solution s is measured by the ratio of its cost c_s with the cost of the global optimum c_{s^*}, ($\lambda = \frac{c_s}{c_{s^*}}$). We say that a neighborhood structure guarantees ratio ε or that it is ε-approximate, if for every instance x and locally optimal solution $\hat{s}$, the cost of $\hat{s}$ is at most greater than a factor ε of the minimum cost ($\lambda \leq \varepsilon$). Because it must be $\varepsilon \geq 1$, at the maximization problems we have: $\lambda = \frac{c_{s^*}}{c_s}$ with $\lambda \leq \varepsilon$.

3 Some optimization problems and their usual neighborhoods

The following optimization problems are NP-complete and their corresponding local search problems with some common neighborhoods are shown to be PLS-complete. Here we will give the definitions of the problems and some of those neighborhoods.

3.1 Graph Partitioning

Given a graph $G = (V, E)$ with $2n$ nodes and weights $w_e \in \mathbb{Z}^+$ on every edge, find a partition of the set of nodes V in two sets A, B where $|A| = |B| = n$, such that the cost of the cut $w(A, B)$ is minimized. (The maximization version of graph partitioning is equivalent to the minimization version both as optimization and as local search problems under all the following neighborhoods, appropriately modified, [4])

A very simple neighborhood is *Swap* and, as we previously saw, is defined as follows: a partition (A, B) of the set of nodes V into two equal parts has as neighbors all the partitions, which can be produced by swapping a node in A with a node in B.

A much more complex neighborhood is *Kernighan-Lin*. The heuristic that explores this neighborhood moves from one partition to a neighboring one through a sequence of greedy swaps. At each step of the sequence we choose to swap the best pair of nodes among those that have not been moved in previous steps of the sequence. By the term "best" we imply that the swap produces the minimum cost differential, i.e. the weight of the cut decreases the most or increases the least. The opposite holds for maximization problems. In case of a tie, a tie-breaking rule is used to chose only one pair. Thus, a sequence of partitions $(A_i, B_i), i = 1, \ldots, n$ from a partition (A, B) is formed, where $|A_i - A| = |B_i - B| = i$. All these partitions are neighbors of the initial one.

Obviously this neighborhood is more powerful than the simple Swap. Observe that if a partition is a local optimum under Kernighan-Lin then it will be a local optimum under

Swap too, because in the opposite case its first neighbor would be better. But as we will see, both problems are polynomially equivalent.

Another neighborhood, usually used with this problem is *Fiduccia-Mattheyses (FM)*. FM is like Kernighan-Lin with the only difference that each step of the sequence consists of two substeps. In the first substep we move the "best" unmoved node from one side to the other and in the second substep we move the "best" unmoved node of the opposite side to have a balanced partition again. *FM-Swap* is one more neighborhood, obtained from the FM if we use only the first step of FM's sequence of steps.

3.2 Travelling Salesperson Problem

In TSP we are given a complete graph of n nodes ('cities') with positive integer weights ('distances') w_e on each edge and we want to find a least-weight tour that passes exactly once through each city or equivalently we are looking for a simple circle of length n with minimum weight. If the graph is not complete then it might not have a circle of length n. The *Hamiltonian cycle* problem is to find if a graph with n nodes has a simple circle of length n and it is NP-complete. The condition under which at least one of the Hamiltonian cycles is always a cycle of minimum weight, is the weights to satisfy the triangular inequality (i.e. $w_{ij} \leq w_{ik} + w_{kj}$). Then the problem is called *metric TSP*. A more specific case is when the cities are points on the plane and the weights of the edges are their Euclidean distances. If the weights of the edges are not symmetric then we have a directed graph and the problem is called *Asymmetric TSP*. All these problems are NP-complete.

Considering the initial TSP, we can define a whole set of neighborhoods called *k-Opt*, with $k \geq 2$ in general. The neighbors from these neighborhoods are obtained as follows. Starting from an arbitrary Hamiltonian cycle we delete k edges in order to obtain k non-connected paths. Then we reconnect these k paths such that a new tour is produced. The locally optimal solution is then called k-Opt solution. This neighborhoods can be used both on symmetric and asymmetric problems. As we will see later, TSP/k-Opt is PLS-complete for a fixed k, but we don't know anything about small values of k such as in the cases of TSP/2-Opt and TSP/3-Opt.

For the TSP problem there is a neighborhood that permits the replacement of an arbitrary number of edges between two neighboring tours using a greedy criterion to stop and is called *Lin-Kernighan*. The main idea is as follows: Given a tour we delete an edge (a, b) to obtain a Hamiltonian path with ends a and b. Let a be stable and b variable. If we add an edge (b, c) from the variable end, then a circle is created. There is a unique edge (c, d) that incidents on c, whose deletion breaks the circle, producing a new Hamiltonian path with a new variable end d. This procedure is called rotation. We can always close a tour by adding an edge between the stable end, a, and the variable end, d. Thus, a movement of the Lin-Kernighan heuristic from a tour to a neighboring one consists of a deletion of an edge, a greedy number of rotations and then the connection of the two ends. There are a lot of variations of this main schema depending on how we choose a rotation. Such a variation is the *LK'* neighborhood under which the TSP has been shown to be PLS-complete, [5]. It is open if the TSP/Lin-Kernighan is PLS-complete or not.

3.3 Max-Cut

Given an undirected graph $G = (V, E)$ with positive weights on its edges, find a partition of the set of nodes V into two (not necessarily equal) sets, whose cut $w = (A, B)$ has the maximum cost. The version of minimization (Min-Cut) can be solved in polynomial time while the Max-Cut is NP-complete. The simplest neighborhood for Max-Cut is Flip, with which two solutions (partitions) are neighbors if the one can be obtained from the other by moving one node from the one side of the partition to the other. A Kernighan-Lin neighborhood can be defined for this problem, too, where the sequence of steps is a sequence of flips of nodes.

3.4 Max-Sat

In maximum satisfiability *(Max-Sat)* we have a boolean formula in conjunctive normal form (CNF) with a positive integer weight for each clause. A solution is an assignment of 0 or 1 to all the variables. Its cost, to be maximized, is the sum of the weights of the clauses that are satisfied by the assignment. *Max-k-Sat* is the same problem with the restriction of at most (or sometimes exactly) k literals in each clause. The simplest neighborhood for this problem is also the Flip neighborhood, where two solutions are neighbors if one can be obtained from the other by flipping the value of one variable. A Kernighan-Lin neighborhood can be defined for this problem, too, where the sequence of steps is a sequence of flips of variables.

3.5 Not-all-equal Max-Sat

An instance of not-all-equal maximum satisfiability (*NAE Max-Sat*) consists of clauses of the form NAE($\alpha_1, \ldots, \alpha_k$), where each α_i is either a literal or a constant (0 or 1). Such clauses are satisfied if their elements do not all have the same value. Each clause is assigned a positive integer weight. A solution is again an assignment of 0 or 1 to all the variables and its cost, to be maximized, is the sum of the weights of the satisfied clauses. If we restrict the clauses to have at most (or sometimes exactly) k literals then we have the *NAE Max-k-Sat* problem. The restriction to instances with no negative literals in their clauses is called *Pos NAE Max-Sat*. We can define the Flip neighborhood and the Kernighan-Lin neighborhoods for this problem as for Max-Sat.

3.6 Stable configurations in neural networks of Hopfield type models

We are given a non-directed graph $G = (V, E)$ with a positive or negative weight w_e on each edge e and a threshold t_v for each node v (we can assume that the missing edges have weight equal to 0). A configuration assigns to each node v a state s_v, which is either 1 ('on') or -1 ('off'). These values can also be 1 and 0 but both versions are equivalent. A node is "happy" if $s_v = 1$ and $\sum_u w_{(u,v)} s_u s_v + t_v \geq 0$ or $s_v = -1$ and $\sum_u w_{(u,v)} s_u s_v + t_v \leq 0$. A configuration is stable if all the nodes are happy. The problem is to find a stable configuration for a given network. It is not obvious, a priori, that such a configuration exists. Actually, at the case of the directed graphs it is possible not to exist.

Hopfield, [6], showed that in the case of the undirected graphs always exists such a configuration. To prove this, he introduced a cost function $\sum_{(u,v)\in E} w_{(u,v)}\ s_u s_v + \sum_{v\in V} t_v s_v$ and argued that if a node is unhappy then changing its state will crease the cost. This means that the stable configuration coincide with the local optimum for this function under this "Flip" neighborhood, and a local optimum, of course, always exists.

If all the edge weights are negative, then the stable configuration problem is equivalent to $s-t$ Max-Cut/Flip, while if all the edges are positive the problem is equivalent to $s-t$ Min-Cut/Flip, (s and t are two nodes that must be on different sides of the partition). Additionally, if all the thresholds are 0 then the stable configuration problem is equivalent to Max-Cut/Flip or Min-Cut/Flip, depending on whether the edge weights are negative or positive, respectively, (see [7, 8, 9]). The $s-t$ Min-Cut and Min-Cut problems can be optimally solved in polynomial time, hence the stable configuration problem of neural nets with positive weights is solved polynomially. For the special case of Max-Cut/Flip for cubic graphs we can find local optima in polynomial time.

4 The PLS class

PLS class was designed to include those Local Search problems which are related to the usual local search heuristics. All these problems have the following properties in common. They find initial solutions, compute solution costs and search a neighborhood "easily", that is in polynomial time.

Typically PLS is defined as follows: Consider a local search problem Π. We assume that its instances are coded in binary strings and for every instance x, its solutions $s \in \mathcal{J}_\Pi(x)$ are binary strings, too, with their length bounded by a polynomial on x's length. Without loss of generality we can, also, assume that all the solutions are coded as strings of the same length $p(|x|)$. Finally, we assume, for simplification, that the costs are non-negative integers (this theory can be straightforwardly expanded to rational costs, too).

Definition 4.1 *A Local Search problem Π is in PLS if there are three polynomial time algorithms A_Π, B_Π, C_Π with the following properties:*

1. *Given a string $x \in \{0,1\}^*$, algorithm A_Π defines if x is an instance of Π and in that case produces a solution $s_0 \in \mathcal{J}_\Pi(x)$.*
2. *Given an instance x and a string s, algorithm B_Π defines if $s \in \mathcal{J}_\Pi(x)$ and if it is so, then it computes the cost $f_\Pi(s,x)$ of the solution s.*
3. *Finally, given an instance x and a solution s, algorithm C_Π defines if s is a local optimum, and if it is not C_Π gives a neighbor $s' \in \mathcal{N}_\Pi(s,x)$ with a strictly better cost, i.e. $f_\Pi(s',x) < f_\Pi(s,x)$ for a minimization problem and $f_\Pi(s',x) > f_\Pi(s,x)$ for a maximization problem.*

All common local search problems, are in PLS. From the definition we can construct a local search algorithm which starts with the initial solution $s_0 = A_\Pi(x)$ and applies iteratively algorithm C_Π until it reaches a local optimum.

Having defined the PLS class, we are wondering where it lies in relation to the other known classes. The largest part of the Complexity Theory relies on the Decision Problems ($|O_\Pi(x)| = 1$, for each x - "YES-NO" Questions). In particular, the fundamental classes P and NP are classes of the Decision Problems. Usually these two classes are sufficient for the determination of the complexity of the Optimization Problems. On the one hand, we can prove that an Optimization Problem is "easy" if we can show a polynomial time algorithm that solves it. On the other hand if the optimization problem is "hard" we can usually show this by transforming it to a related decision problem, OP-decision, and show the latter to be NP-complete.

The OP-decision problem is defined as follows:

> Given an instance x and a cost c, is there any solution $s \in \mathcal{J}_{OP}(x)$ with cost at least as "good" as c? ($f_{OP}(s,x) \leq c$ for minimization problems, $f_{OP}(s,x) \geq c$ for maximization problems)

Obviously the OP-decision problem is not harder than the OP since an algorithm that solves the OP can be used to solve the OP-decision. If the OP-decision problem is NP-complete then the Optimization Problem will be NP-hard. This means that there is an algorithm for an NP-complete problem which uses an algorithm for the optimization problem as a subroutine and takes polynomial time, apart from the running time of the subroutine's calls.

In Local Search problems unfortunately it does not seem to be such a transformation which gives a proper Decision Problem not harder than the initial one. For that reason the classes P and NP of the decision problems cannot characterize the complexity of the local search problems and we should examine them from the beginning as simple Search Problems.

We define classes NP_S and P_S, which are the search analogues of classes NP and P, as follows:

- NP_S is the class of Search Problems (Relations) $\mathbb{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ which are polynomially bounded and polynomially recognizable. Meaning that
  - if $(x, y) \in \mathbb{R}$ then $|y|$ is polynomially bounded in $|x|$ and
  - there is a polynomial time algorithm that given a pair (x, y) determines whether it belongs to $\mathbb{R}$ or not.
- Such a problem $\mathbb{R}$ is in P_S if there is a polynomial time algorithm that solves it, i.e., given an instance x, either produces as output a y such that $(x, y) \in \mathbb{R}$ or it reports (correctly) that there is no such y.

Easily follows from the definitions that

Proposition 4.1 $P_S = NP_S$ *if and only if* $P = NP$

PLS lies somewhere between P_S and NP_S. On the one hand, we see that any problem in P_S can be formulated as a PLS-problem. That is, we can define for each instance x of $\mathbb{R}$ a set of solutions $\mathcal{J}(x)$, a cost function $f(y, x)$ and a neighborhood function $\mathcal{N}(y, x)$ along with the corresponding algorithms A, B, C, which will satisfy the conditions in the definition of the PLS, such that $(x, y) \in \mathbb{R}$ if and only if y is a local optimum for x. Simply, let $\mathcal{J}(x) = y : (x, y) \in \mathbb{R}$ and for each $y \in \mathcal{J}(x)$ let $f(y, x) = 0$ and $\mathcal{N}(y, x) = y$. Algorithm A of definition 4.1 is the polynomial algorithm that solves the problem $\mathbb{R}$, algorithm B uses the algorithm that recognizes the members of $\mathbb{R}$ and algorithm C is trivial.

On the other hand, every problem Π in PLS is also in NP_S The relation $\{(x, y) : y$ is locally optimal for $x\}$ is polynomially recognizable from the algorithm C_Π of definition 4.1. Hence we have the following theorem

Theorem 4.1 $P_S \subseteq PLS \subseteq NP_S$

Observing the previous theorem we are now wondering if we can conclude in a more tight relation. The question now is if PLS coincides with any of its bounds in the above relation or if it is properly between them. On the lower side it is not clear if PLS can be equal to P_S. Although we cannot conclude anything about this using the current computer theory, such a result would be remarkable, since it would require a general method for finding local optima at least as clever as the ellipsoid algorithm for the Linear Programming, which is one of the simplest and with very good behavior member of PLS.

On the upper side we have strong complexity theoretic evidence of proper containment. We know that NP_S includes NP-hard problems. For example the relation that consists of the pairs {x= a graph, y= a Hamilton cycle of x} is in NP_S. It is NP-hard to solve this search problem since it includes the solution of the Hamiltonian cycle problem, which is a decision NP-complete problem. The following fact shows that it is very unlikely that PLS contains NP-hard problems.

Theorem 4.2 *If a PLS problem Π is NP-hard then NP=co-NP.*

Proof. If Π is NP-hard then there is, by definition, an algorithm M for any NP-complete problem X, that calls an algorithm for Π as a subroutine and takes polynomial running time, apart from the time spent during the subroutine calls. But then we can verify that a given string x is a 'no'-instance of X in non-deterministic polynomial time as follows: just guess a computation of M with input x, including the inputs and the outputs of the calls to the subroutine for Π. The validity of the computation of M outside of the subroutines can be checked in deterministic polynomial time , by our assumption of M. The validity of the subroutine outputs can be verified using the polynomial time algorithm C_Π, whose existence is implied by the fact that Π is in PLS, to check whether the output is really a locally optimal solution for the input. Thus the set of 'no'-instances of X is in NP, i.e. $X \in co - NP$. Since X is NP-complete, it is implied that $NP = co - NP$. □

Formerly we saw that we cannot use NP-hardness to relate the Local Search problems to the class NP and argue that they are intractable, as we do with the Optimization problems. Therefore, in order to achieve something similar, we will relate them to each other with proper reductions and we will identify the hardest problems in PLS.

Definition 4.2 *Let Π_1 and Π_2 two Local Search problems. A PLS-reduction from Π_1 to Π_2 consists of two polynomial time computable functions h and g such that:*

1. *h maps instances x of Π_1 to instances $h(x)$ of Π_2*
2. *g maps pairs of the form (solution of $h(x)$,x) to solutions of x and*
3. *for all instances of Π_1, if s is local optimum for the instance $h(x)$ of Π_2 then $g(s, x)$ is a local optimum of x.*

If there is such a reduction, then we say that Π_1 PLS-reduces to Π_2.

Notice that, by its definition, the two requisitions that g satisfies are to map every solution t_i of $h(x)$ to one solution s_j of x, and if t_i is a local optimum of $h(x)$, then $g(t_i, x) = s_j$ is a local optimum of x. So we can argue that through a PLS-reduction we are transferred in an instance of a problem with, probably, fewer solutions and *local optima* than the initial one. The only case where $h(x)$ can have more solutions than x is when more than one of the solutions of the $h(x)$ are mapped to only one solution of x.

It is easy to see that for the PLS-reductions the transitional property holds and that they allow us to relate the difficulty of a problem with that of another one.

Proposition 4.2 *If* Π_1, Π_2 *and* Π_3 *are problems in PLS such that* Π_1 *PLS-reduces to* Π_2 *and* Π_2 *PLS-reduces to* Π_3*, then* Π_1 *PLS-reduces to* Π_3*.*

Proposition 4.3 *If* Π_1 *and* Π_2 *are problems in PLS such that* Π_1 *PLS-reduces to* Π_2 *and if there is a polynomial time algorithm for finding local optima for* Π_2*, then there is also a polynomial-time algorithm for finding local optima for* Π_1*.*

Definition 4.3 *A problem* Π*, which is in PLS, is PLS-complete if every problem in PLS can PLS-reduce to it.*

We will now give the definition of the first problem which is showed in [4] to be PLS-complete. This problem is called *Circuit/Flip*. An instance of this problem is a combinatorial Boolean circuit x (more precisely its encoding) which consists of AND, OR and NOT gates or any other complete Boolean basis. Let x has m inputs and n outputs. The set of the solutions $\mathcal{J}(x)$ consists of all the binary strings of length m, i.e. all the possible inputs. The neighborhood $\mathcal{N}(s, x)$ of a solution s consists of all the binary strings of length m whose Hamming distance from s equals to 1. Remember that two binary strings have Hamming distance equal to one if they are different exactly in one bit. The cost of a solution s is the output vector of the circuit for input s, which expresses a number written in binary. More typically it is $f(s, x) = \sum_{j=1}^{n}(2^{j-1}y_j)$, where y_j is the jth output of the circuit with input s, reading from right to left. The problem can be defined either as a maximization or as a minimization problem, since as we will see soon the two versions are equivalent. Intuitively the local search problem asks for an input such that its output cannot be improved lexicographically by flipping a single input bit.

It is easy to see that Circuit/Flip is in PLS. Let algorithm A returns the vector of length m with all its digits equal to 1. Let algorithm B checks that the given binary string has length m and then computes the circuit x with input s. Finally, let algorithm C that computes the circuit x for all the input vectors with Hamming distance from s equal to 1 (there are only m of them). Algorithm C returns a vector if it has better cost than s. Hence from the Definition 4.1 we have that Circuit/Flip $\in$ PLS.

The maximization and the minimization versions of the Circuit/Flip problem are equivalent both for the local search problem and for the respective optimization problem. In order to convert an instance of one form to one of the other, which will have the same global and local optima, we simply add another level of logic in the circuit, which will flip the value of all the output variables (changes 1's to 0's and vice versa). Indeed, this transformation is a PLS-reduction from the one version to the other. Sometimes we use the prefixes Max- and Min- to clarify which version of Circuit/Flip we are referring to.

Theorem 4.3 *Both the maximization version and the minimization version of the Circuit/Flip problem are PLS-complete.*

The proof is omitted, here (see [4] for the complete proof), however, we will give some hints about it. First, it is showed that any problem L in PLS can be PLS-reduced to an intermediate problem Q, which has the same instances with L but its solutions and neighborhood function are the same with the Circuit/Flip problem. Thus, Q can be straightforwardly PLS-reduced to Circuit/Flip by making a circuit which computes the cost function of Q, with the same inputs and outputs.

We mentioned that Q has the same neighborhood with Circuit/Flip, i.e. the Flip neighborhood. This can be done, since any neighborhood of L, as complex as it is, will simply perturb, in a polynomial way, the bits that consist the encoding of a solution to produce new ones. Hence, all the complexity of the Q problem is shifted in its cost function. Problem Q overcomes the weakness of its very simple neighborhood, because it has three basic characteristics:

1. It corresponds to every solution s of a problem L a solution $ss00$, and a number of intermediate solutions with appropriate costs, such that there is access, through simple flips of one bit at a time, from $ss00$ to the correspondent solution of any perturbation s' of s.
2. It preserves the relative ordering of the solutions of L in respect to their costs. Hence, when a solution s of L has bigger (smaller) cost of another solution s', then all the corresponding solutions of s in Q will have bigger (smaller) costs from the corresponding solutions of s'.
3. Algorithm's B of Q definition is based on the algorithms B and C of L. Thus, the transition of Q's heuristic, from a group of solutions to a neighboring one, will be executed if and only if the L's heuristic would do the corresponding transition.

So, the Q problem has the same local and global optima with L, and their heuristics follow similar paths. Therefore the only differences between an instance of L and an instance of Q are the following:

- The costs of the solutions of Q are bigger, by a constant factor, from those of the corresponding solutions of L.
- Two neighboring solutions in L are not directly connected in Q, but through a unique obligate path. This path consists of a number of intermediate solutions, where each of them differs in one bit from its previous solution.

It has also been proved that

Theorem 4.4 *The following problems are PLS-complete:*

1. *Graph Partitioning under the Kernighan-Lin neighborhood (for every tie-breaking rule), [1], and under the following neighborhoods: (a) Swap, (b) Fidducia-Mattheyses, (c) FM-Swap, [10].*
2. *Travelling Salesman Problem under the k-Opt neighborhood for some fixed k, [11], and under the LK' neighborhood, [5].*
3. *Max-Cut/Flip, [10].*
4. *Max-2Sat/Flip, [12].*
5. *Pos NAE Max-3Sat/Flip, [10].*
6. *Stable configurations for neural networks, [10].*

5 Complexity of the standard local search algorithm

In this section we will be concerned with the running time of local search algorithms (Running Time Problem) and we will see how the theory of PLS-completeness can be adapted to study this issue. At first, we will give some definitions.

Definition 5.1 *Let* Π *a local search problem and let* x *an instance of* Π. *The* neighborhood graph $NG_\Pi(x)$ *of the instance* x *is a directed graph with a node for each feasible solution of* x *and an arc* $s \to t$ *when* $t \in \mathcal{N}_\Pi(s,x)$. *The* transition graph $TG_\Pi(x)$ *is the subgraph which includes all those arcs for which the cost* $f_\Pi(t,x)$ *is strictly better than* $f_\Pi(s,x)$ *(greater if* Π *is a maximization problem or smaller if* Π *is a minimization problem). The* height of a node v *is the length of the shortest path in* $TG_\Pi(x)$ *from* v *to a* sink, *that is a node with no outgoing arcs. The* height of $TG_\Pi(x)$ *is the greatest of the heights of its nodes.*

Since the transition graph expresses the additional information of the cost difference between two neighboring solutions, we can imagine a third dimension for the cost of each solution. Hence, we would obtain the transition graph from the neighborhood graph if we only kept the downward arcs, for a minimization problem, or the upward arcs for a maximization problem.

We will be concerned mainly with the transition graph. Note that $TG_{\Pi}(x)$ is an acyclic graph. Also see that the cost induces a topological ordering of the nodes: the arcs head from worst to better nodes. Hence, the local optima are the sinks of the graph. $TG_{\Pi}(x)$ represents the possible legal moves for a local search algorithm on instance x. Beginning from some node (solution) v, the standard local search algorithm follows a path from node to node until it reaches a sink. The length of that path is the number of the iterations of the algorithm, which determines its running time. The precise path that is been followed (and hence the complexity) is determined by the pivoting rule that we have chosen. At each node which is not a sink the pivoting rule chooses which of the outgoing arcs will be followed. The height of a node v is the lower bound on the number of iterations which are needed by the standard local search algorithm even if it uses at each iteration the best pivoting rule. Note that this rule may not be computable in polynomial time on the size of the instance (in general the transition graph has an exponential number of nodes).

If a local search problem has instances whose transition graph has exponential height, the standard local search algorithm will need exponential time in the worst case, regardless of how it chooses better neighbors. This turns out to be the case with all the problems that have been shown to be PLS-complete. The notion of PLS-reduction that we have defined is not adequate to prove this, but it can be strengthened in an appropriate way.

Definition 5.2 *Let P and Q two local search problems and let (h, g) a PLS-reduction from P to Q. We say that the reduction is* tight *if for every instance x of P we can choose a subset R from the feasible solutions of the image instance $y = h(x)$ of Q so that the following properties are satisfied:*

1. *R includes all local optima of y*
2. *For every solution p of x we can construct in polynomial time a solution $q \in R$ of y such that $g(q, x) = p$*
3. *Suppose that the transition graph of y, $TG_Q(y)$, includes a directed path from $q \in R$ to $q' \in R$, such that all the internal nodes of the path are outside R and let $p = g(q, x)$ and $p' = g(q', x)$ the respective solutions of x. Then either it will hold $p = p'$ or $TG_P(x)$ will include an arc from p to p'.*

See that the tight PLS-reduction is a PLS-reduction, i.e. all the solutions of $h(x)$ have a corresponding solution in x through g and the local optima of $h(x)$ correspond in local optima of x. In addition, there is a subset R in $h(x)$ though, which includes all the local optima of $h(x)$ and all the solutions of x have at least one corresponding solution in R. Hence, x has fewer or equal number of solutions as the subset R of $h(x)$. Notice, also, that since all local optima are in R, all paths of $h(x)$ will end up in there.

The third property tells us that if a path gets out of R, then the solution that reaches when it comes again inside R cannot have smaller distance (in arcs-steps) from the initial than the distance between their corresponding solutions in x. By this restriction we ensure that the solutions outside R are not helpful for the decrease of the complexity of the problem with the local search method. Therefore

Lemma 5.1 *Suppose that P and Q are problems in PLS and that h, g define a tight PLS-reduction from the problem P to the problem Q. If x is an instance of P and $y = h(x)$ is its image in Q, then the height of $TG_Q(y)$ is at least as large as the height of $TG_P(x)$. Hence, if the standard local search algorithm of P takes exponential time in the worst case, then so does the standard algorithm for Q.*

Proof. Let x be an instance of P, let $TG_P(x)$ be its transition graph and let p be a solution (node) whose height is equal to the height of $TG_P(x)$. Let $y = h(x)$ and let $q \in R$ be a solution of y such that $g(q, x) = p$. We claim that the height of q in $TG_Q(y)$ is at least as large as the height of p in $TG_P(x)$. To see this, consider a shortest path from q to a sink of $TG_Q(y)$ and let the nodes of R that appear on this path be $q, q_1, \ldots, q_k$. Let $p_1, \ldots, p_k$ be the images under g of these solutions, i.e., $p_i = g(q_i, x)$. From the definition of a tight reduction, we know that q_k is a local optimum of y, and thus p_k is a local optimum of x. Also, for each i, either $p_i = p_{i+1}$ or there is an arc in $TG_P(x)$ from p_i to p_{i+1}. Therefore, there is a path of length at most k from node p to a sink of $TG_P(x)$. □

It is easy to see that we can compose tight reductions. Tight reductions allow us to transfer lower bounds of the running time of the local search algorithm from one problem to another. All PLS-complete problems that we have referred to are complete under tight reductions.

To prove that in the worst case the running time of the standard local search algorithm for the tightly PLS-complete problems is exponential, it suffices to show that there is a problem in PLS which has such a property.

Lemma 5.2 *There is a local search problem in PLS whose standard local search algorithm takes exponential time.*

Proof. Consider the following artificial minimization problem. For every instance x of size n, the set of solutions consists of all n-bit integers $0, \ldots, 2^n - 1$. For each solution i, its cost is i and if $i > 0$ then it has one neighbor, $i - 1$. Hence, there is a unique local and global optimum, namely 0, and the transition graph is a path from $2^n - 1$ to 0. The local search algorithm, starting at $2^n - 1$, will follow this path and will stop after an exponential number of iterations. □

Theorem 5.1 *The standard local search algorithm takes exponential time, in the worst case, for all the problems referred in Theorem 4.4*

Note that the exponential bounds hold for every pivoting rule, including randomized and non-polynomially computed rules.

Hence we have a general approach for proving bounds on the complexity of the local search heuristics. Outside that, however, there have been very few results, based mostly on ad hoc methods and for particular pivoting rules.

6 The quality of local optima

As we have said, the local search method is usually applied to tackle hard optimization problems. By imposing a neighborhood structure upon the solutions of a problem and by searching a local, only, optimum we achieve to decrease the complexity of the problem. The only restriction in neighborhoods is that they must be searchable efficiently, that is in polynomial time. Ideally we would like to have an exact neighborhood, one in which the global and the local optima coincide. Unfortunately, something like that is rare and as intuitively one would understand, this decrease of complexity has an impact on the quality of the optima that we can guarantee to find. Typically we can say the following:

At first, remember that a problem is called *strongly* NP-hard if it remains NP-hard even when the weights (costs) of its instances are polynomially bounded.

Theorem 6.1 *Let Π an optimization problem and $\mathcal{N}$ a neighborhood function such that the local search problem $\Pi/\mathcal{N}$ is in PLS.*

1. *If Π is strongly NP-hard (respectively NP-hard), then $\mathcal{N}$ cannot be exact unless P=NP (resp. NP=co-NP).*
2. *If the approximation of Π within a factor ε is strongly NP-hard (resp. NP-hard), then $\mathcal{N}$ cannot guarantee ratio ε unless P=NP (resp. NP=co-NP).*

Proof. Let Π be a strongly NP-hard problem and lets consider an instance with polynomially bounded weights. Then the standard local search algorithm will converge in polynomial time. If, furthermore, the neighborhood is exact, then the solution that will be computed, will be a global optimum.

In general suppose that Π is an NP-hard (possibly, not strongly) optimization problem. Let it be a minimization problem. Typically the following decision problem is NP-complete: Given an instance x and a value υ, is there any solution with cost at most υ? If $\mathcal{N}$ is an exact neighborhood, then we can solve its complementary decision problem in non-deterministic polynomial time as follows: Given x and υ, guess a solution $\hat{s}$ and verify that $\hat{s}$ is locally (therefore globally) optimum and that its cost satisfies the relation $f(\hat{s}) > \upsilon$. □

The analogous statements about the approximation ratio follow by the same arguments. All the problems that we have seen until now (TSP, Graph Partitioning, Max-Cut, Max-Sat) are strongly NP-hard.

So, we can't find a global optimum if an optimization problem is NP-hard or ε-approximate it if this approximation is NP-hard, unless NP=co-NP. In the previous section we also saw that the standard local search heuristic takes exponential time to find a local optimum for many interesting problems under a lot of usual neighborhoods (the PLS-complete problems).

A combination of these questions and an even weaker goal would be to guarantee an ε-approximation for any local optimum in polynomial time with the standard local search heuristic. In [3] it is proved that even such a guarantee cannot hold at least for the Circuit/Flip, the Graph Partitioning/KL and any other PLS-complete problem, that is shown complete under a tight and weight-preserving PLS-reduction.

Recently, in [13], Orlin et al introduced the notion of the ε-local optimum to be a solution $\bar{S}$, where $cost(\bar{S}) \leq (1+\varepsilon)cost(S)$, for all $S \in N(\bar{S})$. They, also, presented a "fully polynomial ε-local approximation algorithm", which finds such solutions in $O(n^2\varepsilon^{-1}logn)$.

7 Approximation results

Empirically, it is well known that the local search heuristics seem to produce very good approximate solutions. For example, the heuristic algorithms for TSP ends up to solutions which are very close to the global optimum, in "typical" instances of the problem, on the plane. They are even better than other algorithms which have better approximation performance in the worst case (for instance, the 3/2-approximation algorithm which presented in [14])

During the last years, a lot of local search algorithms were proved to give an ε-approximate solution of the global optimal solution of subcases of many characteristic and popular problems, in a rather competitive running time. This Section presents some results of this nature.

7.1 Max-k-Sat

Max-Sat is the first example of an NP-complete problem. It cannot, also, be in PTAS, unless P=NP. However, randomized local search solves 2-SAT in polynomial time, [15]. There is a lot of research in solving the k-SAT problem with "weakly exponential" algorithms. Recently, in [16], there was presented a deterministic local search method running in time $(2 - \frac{2}{k+1})^n$ up to a polynomial factor. The Max-k-Sat problem is defined, here, as the Max-Sat problem in Section 3.4, without weights, and an additional requirement of *exactly* k literals per clause.

Another set of neighborhoods used with this problem are the d-neighborhoods, meaning that the neighboring assignments have different values on up to d variables. Thus, the 1-neighborhood, also called Flip, is the neighborhood where only one variable of an assignment can be flipped, in order to obtain a neighboring one.

In [17], the following theorem, for this problem under the Flip neighborhood, is proved

Theorem 7.1 *Let m and m_{loc} be the number of satisfied clauses at a global and a local optimum, respectively, of any instance of the unweighted MAX-k-SAT. Then we have $m_{loc} \geq \frac{k}{k+1}m$, and this bound is sharp.*

Proof. Without loss of generality, we can assume that in the local optimum each variable is assigned the value true. If it is not the case, by putting $x'_i = \bar{x}_i$ if $x_i \leftarrow$ false, and $x'_i = x_i$ if $x_i \leftarrow$ true in the local optimum, we obtain an equivalent instance for which the assumption holds.

Let δ_i the variation of the number of satisfied clauses when variable x_i is flipped. Since the assignment is a local optimum, flipping any variable decreases the number of satisfied clauses, i.e. $\delta_i \leq 0$, for $1 \leq i \leq n$.

Let cov_s the subset of clauses that have exactly s literals matched by the current assignment, and $cov_s(l)$ the number of clauses in cov_s that contain literal l.

We have $\delta_i = -cov_1(x_i) + cov_0(\bar{x}_i)$. Indeed, when x_i is flipped from true to false one looses the clauses that contain x_i as the single matched literal, i.e. $cov_1(x_i)$ and gains the clauses that have no matched literal and that contain $\bar{x}_i$, i.e. $cov_0(\bar{x}_i)$.

After summing over all variables, we obtain $\sum_{i=1}^n \delta_i \leq 0$, thus $\sum_{i=1}^n cov_0(\bar{x}_i) \leq \sum_{i=1}^n cov_1(x_i)$. By using the following equality $\sum_{i=1}^n cov_1(x_i) = |cov_1|$ and $\sum_{i=1}^n cov_0(\bar{x}_i) = k|cov_0|$ which can be easily verified, we obtain $k|cov_0| \leq |cov_1| \leq m_{loc}$. Therefore $m = m_{loc} + |cov_0| \leq (1 + \frac{1}{k})m_{loc} = \frac{k+1}{k}m_{loc}$. □

Thus, the local search algorithm with the flip neighborhood is a $\frac{k}{k+1}-$ approximation algorithm for the unweighted MAX-k-SAT. Notice that this algorithm is polynomial, since the problem is unweighted.

A variation of local search is the *non-oblivious* local search, in which the original objective function of the problem is changed in order to guide the algorithm to better local optima. In [18] we have the following theorem.

Theorem 7.2 *The performance ratio*[2] *for any oblivious local search algorithm with a d-neighborhood for MAX-2-SAT is 2/3 for any* $d = O(n)$*. Non-oblivious local search with the flip neighborhood achieves a performance ratio* $1 - \frac{1}{2^k}$ *for MAX-*k*-SAT.*

For Max-2-Sat, for example, the non-oblivious objective function is a weighted linear combination of the number of clauses with one and two matched literals. Namely, $f_{NOB} = \frac{3}{2}|cov_1| + 2|cov_2|$, instead of the oblivious objective function $f_{OB} = |cov_1| + |cov_2|$. It can be shown that the above theorem cannot be improved by using a different weighted linear combination of $|cov_1|$ and $|cov_2|$.

In [19], better approximation algorithms can be obtained by using at first a non-oblivious local search algorithm, and then an oblivious local search algorithm starting with the solution obtained by the first algorithm.

7.2 Max-Cut

Formally, for the *unweighted Max-Cut* problem, we have the following definition: Given a graph $G = (V, E)$, find a partition (V_1, V_2) of V into disjoint sets V_1 and V_2, which maximizes the cardinality of the cut, i.e., the number of the edges with one end point in V_1 and one end point in V_2.

This problem was one of the first problems shown to be NP-complete, but it can be solvable in polynomial time for planar graphs and a few other special cases. It has also been proved that no 0.941-approximation[3] algorithm can exist, unless P=NP, [20].

The best approximation result so far is given from the Goemans and Williamson's algorithm and it is 0.878-approximative. This algorithm reformulates an integer program as a semidefinite program and solves it using a variation of the interior point method for linear programming, [21, 22]. However, it becomes very slow for instances with $n \geq 500$, and because of its complex design it cannot be easily implemented on dedicated circuits.

Considering the unweighted Maximum-Cut under the Flip neighborhood, we have the following theorem (see [23]). Notice that since it is unweighted, the heuristic will converge in polynomial time.

[2] the approximation ratio reversed here

[3] Sometimes, it is used the same definition of the approximation ratio for the maximization problems as that for the minimization (not the inverse), thus ε is smaller than one.

Theorem 7.3 *Given an instance x (a graph $G = (V, E)$) of the Max-Cut problem without weights on its edges, let (V_1, V_2) a locally optimal partition under the Flip neighborhood and let $m_A(x)$ the cost of the locally optimal partition of x under Flip. Then*

$$\frac{m^*(x)}{m_A(x)} \leq 2,$$

where $m^*(x)$ is the cost of the optimal partition.

Proof. Let (V_{1k}, V_{2k}) be the neighbor of a solution (V_1, V_2), by flipping vertex υ_k from the one subset of nodes to the other. Let m the number of the edges of the graph. Since $m^*(x) \leq m$ it suffices to show that $m_A(x) \geq \frac{m}{2}$.

We symbolize with m_1 and m_2 the number of the edges that join the nodes inside V_1 and V_2, respectively, of a local optimum. We have

$$m = m_1 + m_2 + m_A(x). \tag{1}$$

Given any node υ_i, we define

$$m_{1i} = \{\upsilon | \upsilon \in V_1 \text{ and } (\upsilon, \upsilon_i) \in E\}$$

and

$$m_{2i} = \{\upsilon | \upsilon \in V_2 \text{ and } (\upsilon, \upsilon_i) \in E\}$$

Since (V_1, V_2) is a local optimum, then for each node υ_k the solution that comes up from the (V_{1k}, V_{2k}) has as value at most $m_A(x)$. This means that for each node $\upsilon_i \in V_1$,

$$|m_{1i}| - |m_{2i}| \leq 0$$

and for each node $\upsilon_j \in V_2$,

$$|m_{2j}| - |m_{1j}| \leq 0.$$

Summing up all the vertices in V_1 and V_2 we obtain

$$\sum_{\upsilon_i \in V_1} (|m_{1i}| - |m_{2i}|) = 2m_1 - m_A(x) \leq 0$$

and

$$\sum_{\upsilon_j \in V_2} (|m_{2j}| - |m_{1j}|) = 2m_2 - m_A(x) \leq 0.$$

Therefore, $m_1 + m_2 - m_A(x) \leq 0$. From this inequality and from equation 1 it is implied that $m_A(x) \geq m/2$ and the theorem is proved. □

Another algorithm, which uses the main idea of Goemans and Williamson's algorithm in combination with the local search method is LORENA introduced in [24]. It does not have the disadvantages of the Goemans and Williamson's algorithm (time consuming and difficulty on circuit implementation), but is only proved to be 0.39-approximative. Experimental results, though, show a much better approximative performance.

7.3 Travelling Salesperson Problem

The Travelling Salesperson Problem, defined in section 3.2, is a (strongly) NP-hard and PLS-complete problem. It has been shown, in [25], that there are local optima arbitrarily worst than the global optimum. It has also been shown in [26] that finding any polynomial time ε-approximation algorithm for TSP is NP-hard.

When the triangle inequality is present there is a $3/2$-approximation algorithm (see [14]). However the local search heuristic is used in practice, because it is faster and gives good approximation solutions in most cases. We, also, have the following result.

Theorem 7.4 (Chandra 99, [27]) *A local search algorithm with 2-Opt neighborhood achieves a $4\sqrt{n}$ approximation ratio for the* metric TSP.

Proof. Let $T(V)$ be any tour which is locally optimal with respect to the 2-opt neighborhood. Let E_k, for $k \in \{1, \dots, n\}$ the set of big edges of $T(V) : E_k = \{e \in T(V) | wt(e) > \frac{2C_{opt}}{\sqrt{k}}\}$, where C_{opt} is the cost of the global optimum tour. Then the first part of the proof is to show that $|E_k| < k$. Assuming this last result is true, then it means that the weight of the k-th largest edge in $T(V)$ is at most $\frac{2C_{opt}}{\sqrt{k}}$, therefore

$$\begin{aligned} C &= \sum_{k=1}^{n} weight(\text{k-th largest edge}) \\ &\leq 2C_{opt} \sum_{k=1}^{n} \frac{1}{\sqrt{k}} \\ &\leq 2C_{opt} \int_{x=0}^{n} \frac{1}{\sqrt{x}}\, dx \\ &= 4\sqrt{n} C_{opt}. \end{aligned}$$

The proof of $|E_k| < k$ is by contradiction. Here we give only an idea of the proof. Give an orientation of the tour T. Let $t_1, \dots, t_r$, with $r = |E_k| \geq k$ be the tails of each arc from E_k in tour $T(V)$. Then it can be shown that there exists at least $\sqrt{k}$ tails which are at a distance at least $C_{opt}/\sqrt{k}$ from each other. Consider the travelling salesman instance restricted on this set V' of tails. Then the shortest tour on this set has a length $C_{opt}(V')$ greater than $\sqrt{k}\frac{C_{opt}}{\sqrt{k}} = C_{opt}$ contradicting the fact that since the distances satisfy the triangular inequality then for any subset $V' \subseteq V$ one has $C_{opt}(V') \leq C_{opt}(V)$. □

This bound is tight to within a factor of 16. The above theorem combined with the result, also proved in [27], that for random Euclidean instances in the unit square, the expected number of iterations required by 2-Opt is $O(n^{10} logn)$, provides a proof of the quality of local search on such instances. However, for infinitely many n, the k-Opt algorithm can have a performance ratio that is *at least* $\frac{1}{4}n^{\frac{1}{2k}}$.

There is another case, called *TSP(1,2)*, in which every edge can have weight either one or two.

Theorem 7.5 (Khanna 94, [18]) *A local search algorithm with the 2-Opt neighborhood achieves a* $3/2$*-approximation ratio for TSP(1,2).*

Proof. Let $C = v_{\pi_1}, v_{\pi_2}, \ldots, v_{\pi_n, v_{\pi_1}}$ be a local optimum solution with the 2-opt neighborhood. Let O be any optimal solution. To each unit cost edge e in O we associate a unit cost edge e' in C as follows. Let $e = (v_{\pi_i}, v_{\pi_j})$ with $i < j$. If $j = i + 1$ then $e' = e$. Otherwise e' is a unit cost edge among $e_1 = (v_{\pi_i}, v_{\pi_{i+1}})$ and $e_2 = (v_{\pi_j}, v_{\pi_{j+1}})$. Indeed, either e_1 or e_2 must be of unit cost. If it is not the case, then the tour C', obtained from C by removing edges e_1 and e_2 and adding edges e and $f = (v_{\pi_{i+1}}, v_{\pi_{j+1}})$, has a cost at least one less than C and therefore C would not be a local optimal solution with the 2-opt neighborhood.

Let U_O denotes the set of unit cost edges in O and U_C the set of unit cost edges in C obtained from U_O using the above mapping. Since an edge $e' = (v_{\pi_i}, v_{\pi_{i+1}})$ in U_C can only be the image of unit cost edges incident on v_{π_i} in O and since O is a tour, there are at most two edges in U_O which map to e'. Thus $|U_C| \geq |U_O|/2$ and we obtain $\frac{cost(C)}{cost(O)} \leq \frac{|U_O|/2+2(n-|U_O|/2)}{|U_O|+2(n-|U_O|)} \leq \frac{3}{2}$. □

The above bound is shown to be asymptotically tight in [18].

7.4 Other Graph Problems

Let $G = (V, E)$ be an unweighted graph. A set system (or hypergraph) (S, C) consists of a base set S and a collection C of subsets (or hyperedges) of S. A k-set system is a set system where each set in C is of size at most k. We can also assign weights to the sets of C, as well as to the edges of G, in order to obtain a weighted set system or a weighted graph, respectively.

Now we can consider the following collection of problems:

- **3-Dimensional Matching** Given sets W, X, Y and a set $M \subseteq W \times X \times Y$, find a minimum cardinality matching, i.e. a subset $M' \subseteq M$ such that no two elements of M' agree in any coordinate.
- **k-Set Packing** Given a k-set system (S, C), find a maximum cardinality collection of disjoint sets in C. In a weighted set system we are looking for a maximum cost collection of disjoint sets in C.

- **Maximum Independent Set** Given a graph G, find a maximum cardinality subset of mutually non-adjacent vertices, i.e. a subset $V' \subseteq V$ such that $\upsilon_i, \upsilon_j \in V'$ implies $(\upsilon_i, \upsilon_j) \notin E$. In its weighted case (w-MIS), there are weights assigned to the nodes and we want a maximum weight subset of non-adjacent vertices.

- **Vertex Cover** Given a graph G find a minimum cardinality subset $V' \subseteq V$ such that every edge has at least one endpoint in V'.

- **k-Set Cover** Given a k-set system (S, C) find a minimum cover of S, i.e. a subset $C' \subseteq C$ of minimum cardinality such that every element of S belongs to at least one member of C'.

- **Color Saving** (or Graph Coloring) Given a graph G find an assignment of minimum number of colors to the vertices such that adjacent vertices are of different colors. The objective function is to minimize the total number of vertices minus the total number of colors used.

All these problems are NP-hard and MAX SNP-hard, in general. k-Set Packing is a generalization of Maximum Matching from sets of size two (i.e. edges) to sets of size $1, 2, \ldots, k$, hence, for $k = 2$, it is polynomially solvable even for its weighted case. Also, an Independent Set in the intersection graph $H(S, C)$ corresponds to a k-Set Packing in (S, C). Recall that the *intersection graph* $H(S, C)$ of a hypergraph (S, C) has a vertex for each hyperedge with two hyperedges adjacent if and only if they intersect (as sets). Note that the intersection graph of a k-set system C contains no $k + 1$-*claw*, i.e. no $k + 1$-independent set in the neighborhood of any vertex.

The k-Set Cover problem can be solved in polynomial time by matching techniques, for $k = 2$. For the general case, there is a simple greedy algorithm who has performance ratio $\mathcal{H}_k = \sum_{i=1}^{k} \frac{1}{i}$.

The *k-independent set system* of a graph is the collection of all sets of up to k independent vertices in G. An optimal Set Cover of the independent set system of a graph corresponds to an optimal Color Saving, with objective function just the number of colors used, but the size of the instance might have an exponential blowup.

The usual local search method for the unweighted cases of the above problems consists of t-improvements. That is, at each step, s new items are added in a solution, of a maximization problem, and at most $s - 1$ items are removed from it, for some $s \leq t$. For minimization problems holds the reverse.

The results that follow are obtained with some variation of the above local search neighborhood and are tight. In [28], there are similar results for more graph problems. So we have the following approximation ratios:

- $k/2+\varepsilon$ for Maximum Independent Set problem in $k+1$-claw free graphs when $k\geq 4$, in time $O(n^{log_k 1/\varepsilon})$, and 5/3 for $k=3$. This also applies to k-Set Packing and k-Dimensional Matching, [28].
- $(\Delta+2+1/3)/4+\varepsilon$ for Maximum Independent Set in graphs of maximum degree Δ, in time $\Delta^{O(\Delta log 1/\varepsilon)}n$, [28]. For $\Delta\geq 10$ there is also proved in [18], that an algorithm which outputs the larger solution of those computed by a local search and a greedy algorithm has performance ratio $(\sqrt{8\Delta^2+4\Delta+1}-2\Delta+1)/2$.
- $\frac{2}{3}(d+1)$ for the w-MIS on d-claw free graphs, [29, 30], and $d/2$ with non-oblivious local search, [31].
- $2-2f_s(k)$ for Vertex Cover in $k+1$-claw free graphs, for $k\geq 6$, where $f_s(k)=(klnk-k+1)/(k-1)^2=2-(logk)/k(1+o(1))$. For $k=4,5$ a ratio of 1.5 holds, [28].
- 4/3 for 3-Set Cover. For k-Set Cover, using a half greedy half local search algorithm, a ratio $\mathcal{H}_5-5/12$ holds, [32].
- 1.4 for Color Saving, [28]. For graphs with maximum independent sets of size 3 the performance ratio is 6/5, [32].

Another problem, well approximated with local search, is the minimum VFES problem, which is described below.

7.4.1 Minimum Vertex Feedback Edge Set

The graph-theoretic problem minimum VFES (Vertex Feedback Edge Set) is NP-hard and MAX SNP-hard. However, It is very useful in placing pressure meters in fluid networks or in any other system, formulated as a network, in which Kirchoff's laws are valid and a bijective relation exists between the flow and effort variables, like circuits and electrical networks.

We define the minimum VFES problem as follows: Given a graph, find a feedback edge set incident upon the minimum number of vertices. A feedback edge set is a subset of edges in a graph, whose deletion from the graph make the graph acyclic.

In [33], a $2+\varepsilon$ approximation ratio is obtained, in $O(n^{O(1/\varepsilon)})$, by a local search algorithm is introduced, which can be made very efficient by restricting the neighborhoods to be searched, that is in $O(n^3+n^2f(\frac{1}{\varepsilon}))$, where f is an exponential function, but has no dependence on n. There is also presented a PTAS for the case of planar graphs. The neighborhood used by the algorithm is called k-Local Improvement. The current Feedback Edge Set (FES) has as neighbors all the FESs obtained by a replacement of at most

$k-1$ edges from the FES. The cost of a FES is the number of vertices incident to its edges.

7.5 Classification Problems with pairwise relationships

Generally, a classification problem consists of a set P of *objects* to be classified and a set L of *labels* (the classes). The goal is to assign a label to each object in a way that is consistent with some "observed data" that we have about the problem. Here we are interested about problems whose "observed data" are some pairwise relationships among the objects to be classified. These problems have been studied a lot since they are very useful in areas such as statistics, image processing, biometry, language modelling and categorization of hypertext documents.

A characteristic example, from image processing, is the *image restoration problem*. Consider a large grid of pixels. Each pixel has an "observed" intensity and a "true" intensity that we are trying to determine, since it was corrupted by noise. We would like to find the best way to label each pixel with a (true) intensity value, based on the observed intensities. Our determination of the "best" intensity is based on the trade-off between two competing influences: We would like to give each pixel an intensity close to what we have observed and - since real images are mainly smooth, with occasional boundary regions of sharp discontinuity - we would like spatially neighboring pixels to receive similar intensity values.

To be more precise we give the *Metric Labeling Problem* as defined by Kleinberg and Tardos in [34]. Consider a set P of n objects that we wish to classify and a set L of k possible labels. A *labeling* of P over L is simply a function $f : P \to L$. We choose a label for each object. The quality of our labeling is based on the contribution of two sets of terms

- For each object $p \in P$ and label $i \in L$, we have a non-negative *assignment* cost $c(p,i)$ associated with assigning the label i to the object p.
- We have a graph G over the vertex set P, with edge set E indicating the pairwise relationships among the objects. Each edge $e = \{p,q\}$ has a non-negative weight w_e, indicating the strength of this relation.

Moreover, we impose a distance $d(\cdot,\cdot)$ on the set L of labels. So if we assign label i to object p and label j to object q and $e = \{p,q\}$ is an edge of G, then we pay a *separation* cost $w_e d(i,j)$. Thus, the *total cost* of a discrete labeling f is given by

$$Q(F) = \sum_{p \in P} c(p, f(p)) + \sum_{e=\{p,q\} \in E} w_e d(f(p), f(q)).$$

The *labeling problem* asks for a discrete labeling of minimum total cost. Recall that a distance $d : L \times L \rightarrow \mathbb{R}^+$ is a symmetric function and $d(i,i) = 0$ for all $i \in L$. So, if d also satisfies the triangle inequality then d is a *metric*. Hence, the labeling problem is called metric labeling problem if the distance function $d(\cdot,\cdot)$ is a metric on the label set L. Two special cases of the metric d are the *uniform* metric, where $d(i,j) = 1$, for all $i \neq j$, and the *linear* metric, where $d(i,j) = |i - j|, i, j \in \mathbb{N}$. In fact, we can assume, without loss of generality, that the labels of the linear metric are integers $1, 2, \ldots, k$, since in the opposite case we can add the "missing" intermediate integers to the label set and set the cost of assigning them to any vertex to be infinite.

Considering the image restoration problem, with the pixels and their intensities, we can say that

- the assignment cost is getting bigger as the labels we examine become more different than the observed one, for a specific object p,

- the nodes of the graph G are the pixels and there are edges only between neighboring pixels, all with weight equal to 1,

- the distance function d indicates less similarity between two labels and is used to penalize different colors to adjacent pixels. We could simply choose the linear metric to distinguish labels in a grey-scale image, since the color values are integers, but this would lead to over-smoothness of the image and the object boundaries may become fuzzy. So, we would like a non-uniform robust metric, which will sufficiently penalize small differences in the color of neighboring pixels but after a value M, for which we are sure that it is an object boundary, the metric should give the same penalty. Hence, we use the *truncated linear metric* defined as $d(i,j) = min\{M, |i - j|\}$.

In [34], the metric labeling problem is related with other known and well studied problems. So, this problem can be viewed as an extension of the multi-way cut problem, in which we are given a weighted graph with k terminals and we must find a partition of the graph into k sets so that each terminal is in a separate set and the total weight of the edges cut is as small as possible. In the latter problem, there are the terminals which must receive a certain label while all the others do not care of what label they will get, so it is a special case of the metric labeling problem.

It can also be viewed as the *uncapacitated quadratic assignment problem*. In the quadratic assignment problem one must find a matching between a set of n given activities to n locations in a metric space so as to minimize a sum of assignment costs for activities and flow costs for activities that "interact". The metric labeling problem can be obtained from the quadratic assignment by dropping the requirement that at most one activity can be sited at a given location. The activities then correspond to objects and the locations to labels, in the metric labeling problem.

Finally, there is a relation with a general class of Markov random fields. For a given set of objects P and labels L, the *random field* assigns for every labeling f, a probability PR[f]. The random field is *Markovian* if the conditional probability of the label assignment at object p depends only on the label assignments at the neighbors of p in G. If, additionally, the Markov Random Field satisfies the properties of *pairwise interactions* and "metric" *homogeneity,* then is called *metric Markov random field* (see [34] for more details). It is proved that the optimum of a metric labeling problem is equivalent to the optimal configuration of metric Markov random fields, but the transformation is not approximation preserving.

Although the metric labeling problem is NP-hard and MAX SNP-hard, there cases that can be solved polynomially. The cases of $l = 2$ labels (see [35, 36]) and that of the linear metric (see [37, 38, 39]) can be polynomially solved as two-terminal minimum cut problems. Also, Karzanof in [40, 41] showed some other special cases of the labeling problem to be polynomial. Boykof et al. [37] developed a direct reduction from labelings with uniform labelings to multiway cuts, but the reduction is not approximation preserving.

The approximability results obtained for the metric labeling problem and its subcases are the following:

- $O(log|L|loglog|L|)$, for general metrics, [34, 42]
- 2, for the uniform metric, [34, 42]
- 1, for the linear metric and distances on the line defined by convex functions (not necessarily metrics), [42]
- $2 + \sqrt{2} \simeq 3.414$ for the truncated linear metric, [42].

In [42], there is a $O(\sqrt{M})$-approximation result for the *truncated quadratic* distance function ($d(i, j) = min\{M, |i - j|^2\}$), also used in image restoration applications, which is not a metric function. There is also another result which essentially allows us to eliminate the label assignment cost function. That is, there is a reduction from the case with arbitrary assignment costs $c(p, i)$ to the case where $c(p, i) \in \{0, \infty\}$ for all p and i. The reduction preserves the graph G and the optimal solution, but increases the size of the label space from k to nk labels.

All the previous results are obtained by solving the relaxed version of an integer linear program, then rounding its solutions and measuring the gap between them and the solutions of the integer linear program. However, the linear programs involved are quite large and this causes lots of these methods to be too slow and thus less practical. Here, we will present a local search method, which was showed by Gupta and Tardos in [43] to be *4-approximative of the truncated linear metric,* in polynomial time.

7.5.1 A 4-approximation local search method for the metric labeling problem with the truncated linear metric.

Recall that the truncated linear metric is defined as $d(i, j) = min\{M, |i - j|\}$, where i and j are from the set of labels $L = 1, 2, \ldots, l$. In a single local step we consider an interval I of labels of length at most M, and allow any subset of vertices to be related by any of the labels in I. Given a labeling f, we call another labeling f' a *local relabeling* if it can be obtained from f by a local move, i.e., if for all objects $f(p) \neq f'(p)$ implies that $f'(p) \in I$. Unfortunately, we are not able to find the best possible such local move because, as you can see, the neighborhood of a labeling is exponential. However it will be showed later that if the current labeling has cost sufficiently far above the minimum possible cost, then this method will find a move that significantly decreases the cost of the labeling.

In the algorithm we repeatedly pick a random interval I and try to relabel some subset of objects with labels from I, in order to decrease the cost of our labeling. After this local step, each object will either have its label unchanged or will have a label in the interval I. To perform this relabeling efficiently we will create a flow network and find a minimum s-t cut in it. This minimum cut can be associated with a new labeling f' and if f' has a lower cost than the cost of f, we move to the new labeling. In summary the algorithm is the following:

```
Algorithm Local Search
  repeat
    pick a random interval $I$
    build the flow network $N_I$ associated with $I$
    if labeling given by the minimum cut
      on $N_I$ has lower cost
     then move to new labeling
   until a local optimum is reached.
```

The random intervals will be picked in the following manner: we pick a random integer $-M < r < l$, and set I to be the part of the interval of length M starting from offset r that lies in L, i.e. $I = \{r + 1, r + 2, \ldots, r + M\} \cap \{1, 2, \ldots, l\}$. Thus, we have a partition S_r of the label set with at most $\lceil l/M \rceil + 1$ intervals I in it, each of them with size exactly M, except from the initial and the final portion of the line, whose lengths might be smaller than M. Note that the probability, for any pair of labels $i, j \in L$, to lie in different intervals of the partition S_r is exactly $d(i, j)/M$. This algorithm can be

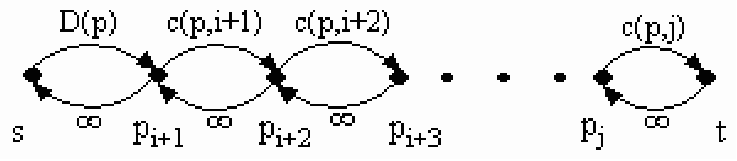


Figure 1: The chain for vertex p

trivially derandomized at a cost of a factor $(l+M)$ increase in the running time. This can be done by considering all possible $(l+M)$ intervals and, for instance, making the moves corresponding to the best possible interval.

The description of the flow network associated with an interval I to which the labels can be changed, follows. Let us consider that the labels in I are $\{i+1, i+2, \ldots, j\}$, with $(j-i) \leq M$. The flow network $N_I = (V, A)$ associated with I is a directed graph with a source s and a sink t, and with capacities on the edges. The first step, to construct it, is for each vertex p of the original graph G to add $(j-i)$ nodes, namely $\{p_{i+1}, \ldots, p_j\}$ to N_I (see Figure 1). We add directed edges (p_k, p_{k+1}) with capacity equal to the assignment cost $c(p,k)$, and directed edges (p_{k+1}, p_k) with infinite capacity, for $i+1 \leq k \leq j$, where $p_{j+1} = t$. Finally, the edge (p_{i+1}, s) is assigned an infinite capacity, while (s, p_{i+1}) is assigned a capacity $D(p)$ which is defined as follows: if $f(p) \in I$ then $D(p) = \infty$ else $D(p) = c(p, f(p))$.

This construction captures the assignment cost. To see this, consider any minimum s-t cut in N_I. The infinite capacity edges ensure that this cut will include exactly one edge from the chain corresponding to each vertex p. If edge (p_k, p_{k+1}) is cut this means that the vertex p is assigned the label k, unless (p_s, p_{i+1}) is cut and $f(p) \notin I$, where the original label is retained for vertex p. Hence, the assignment cost is exactly the capacity of the edge in the cut.

The second step of the construction is to model the separation cost. Let $e = \{p, q\}$ be an edge of the original graph. Depending on the labels of the vertices p, q we have the following cases: 1) If both $f(p)$ and $f(q)$ are not in I, then for each of the corresponding nodes p_k and $q_k, i+1 < k \leq j$, we add a pair of oppositely directed edges between them, each with capacity w_e. We also add a new node υ_{pq} and connect it with the nodes p_{i+1}, q_{i+1} with oppositely directed edges with capacities $w_e d(f(p), i+1)$ and $w_e d(f(q), i+1)$, respectively. Finally, we add an edge (s, υ_{pq}) with capacity $w_e d(f(p), f(q))$, (see Figure 2). 2) If both $f(p)$ and $f(q)$ are in I, then we do nothing. 3) If $f(p) \notin I$ but $f(q) \in I$, then we add an edge (p_{i+1}, q_{i+1}) with capacity $w_e d(f(p), i+1)$.

This structure captures the separation costs. Let f' be the labeling corresponding to the cut, and let us focus on the edge $e = \{p, q\}$. If both vertices retain their original labels, then the cut will be minimized when it passes through (s, υ_{pq}), and incurs a cost of

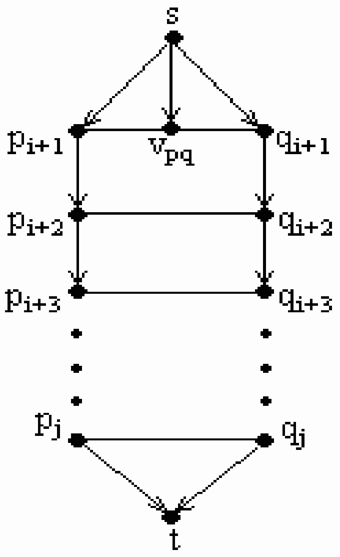


Figure 2: Construction for the edge $\{p, q\}$

$w_e d(f(p), f(q))$. If both vertices are labeled with labels in I, then the cut will be exactly $w_e|f'(p) - f'(q)|$. For the above cases, the cuts equal exactly the separation costs. If one of the vertices (say p) retains its label $f(p) \notin I$ and q is labeled with a new label in I, $f'(q) = k \in I$, then the cut will incur a separation cost $w_e[d(f(p), i+1) + (k - (i+1))]$, which possibly overestimates the actual separation cost in the new labeling.

Note that a minimum cut in N_I can contain at most one of the edges connected with v_{pq}, since only one edge in any of the two pairs of opposite edges can be in the cut, and that in any set of up to three permissible edges, the cost of any two edges is more than the cost of the third one, because $d(\cdot, \cdot)$ satisfies the triangular inequality. A cut is called *simple* if it has finite capacity and it does not cut more than one of the above five edges associated with any edge $e \in E$.

Theorem 7.6 *The simple cuts in the flow network N_I are in one-to-one correspondence with local relabelings f'. The cost of the relabeling $Q(f')$ is no more than the cost of the associated cut, and the cost of the cut overestimates the cost of the labeling by replacing the separation cost $w_e d(f'(p), f(p))$ for edges $e = \{p, q\}$ where exactly one end receives a label in I by a possibly larger term $w_e[d(f'(p), i+1) + d(i+1, f'(q))]$. Further, we have that*

$$d(f'(p), f'(q)) \leq d(f'(p), i+1) + d(i+1, f'(q)) \leq 2M. \tag{2}$$

We will now sketch the proof of the 4-approximability of this method. First we give some definitions. Let f^* be a fixed optimal labeling and let the algorithm's current labeling be f. For any subset $X \subseteq P$, let $A^*(X)$ and $A(X)$ be the assignment cost that the optimum and the current labeling pay respectively for the vertices in X, and for a set of edges $Y \subseteq E$, let $S^*(Y)$ and $S(Y)$ be the separation cost for those edges paid by the optimum and the current solution respectively. So, $Q(f) = A(P) + S(E)$ and the optimum $Q(f^*) = A^*(P) + S^*(E)$.

Consider the case when the algorithm chooses an interval I. Let P_I be the set of vertices of G to which f^* assigns labels from the interval I. Let E_I be the set of edges in $E(G)$ such that the f^*-labels of both endpoints lie in I. Let ϑ_I^- be the set of edges such that exactly one end of the edge has f^*-label in I, the end with higher f^*-label, and ϑ_I^+ be the set of edges that only the end with lower f^*-label has an f^*-label in I. In the proof it is considered a random partition S_r of the labels. Clearly, $P = \cup_{I \in S_r} P_I$ and $\cup_{I \in S_r} \vartheta_I^- = \cup_{I \in S_r} \vartheta_I^+$. ϑ_r is used to denote this union of the *boundary edges* in the partition. Also note that $E = \vartheta_r \cup (\cup_{I \in S_r} E_I)$.

The following lemmas are used in the proof of theorem 7.7. For their proofs see [43].

Lemma 7.1 *For a random partition S_r, the expected value of $M \sum_{e \in \vartheta_r} w_e$ is $S^*(E)$.*

Lemma 7.2 *For a labeling f, and an interval I, the local relabeling move that corresponds to the minimum cut in N_I decreases the cost of the solution by at least*

$$(A(P_I) + S(E_I \cup \vartheta_I^- \cup \vartheta_I^+)) - (A^*(P_I) + S^*(E_I \cup \vartheta_I^-) + M \sum_{e \in \vartheta_I^-} w_e + 2M \sum_{e \in \vartheta_I^+} w_e).$$

Theorem 7.7 *If the labeling f is a local optimum, its cost $Q(f)$ is at most 4 times the optimal cost $Q(f^*)$.*

Proof. The fact that f is a local optimum implies that the improvement indicated by the lemma 7.2 is non-positive for any interval I, i.e., for all I

$$A(P_I) + S(E_I \cup \vartheta_I^- \cup \vartheta_I^+) \leq A^*(P_I) + S^*(E_I \cup \vartheta_I^-) + M \sum_{e \in \vartheta_I^-} w_e + 2M \sum_{e \in \vartheta_I^+} w_e. \quad (3)$$

Now consider a partition S_r and sum these inequalities for each interval $I \in S_r$. On the left hand side we get $A(P) + S(E) + S(\vartheta_r)$ as edges in ϑ_r occur in the boundary of two intervals. This is at least $Q(f)$, the cost of the labeling f. Summing the right hand side, we get exactly $A^*(P) + S^*(E) + 3M \sum_{e \in \vartheta_r} w_e$. So we have that

$$Q(f) \leq A^*(P) + S^*(E) + 3M \sum_{e \in \vartheta_r} w_e$$

for any partition S_r. Taking expectations, the left side is a constant and by lemma 7.1, the expected value of the right hand side is at most $A^*(P) + 4S^*(E)$. Thus we get $Q(f) \leq A^*(P) + 4S^*(E) \leq 4Q(f^*)$. □

7.6 k-Median and Facility Location Problems

There are a lot of different versions of the k-median and facility location problems. To give a general framework for these problems we have to follow a top-down procedure, adding each time the specific requirements that each variation has. There is also the k-means problem, which is very similar to the k-median problem, and it is defined at the end of this section.

Generally, let $N = \{1, 2, \ldots, n\}$ be a subset of *locations* and $F \subseteq N$ be a set of locations at which we may *open* a *facility*. Each location $j \in N$ has a demand d_j that must be shipped to j. For any two locations i and j, let c_{ij} denote *the cost of shipping a unit of demand* from i to j. In these problems the goal is to identify a set of open facilities $S \subseteq F$ and an assignment of locations to S, such that some *cost function* is minimized. The cases where all the unit shipping costs are assumed to be nonnegative, symmetric and satisfy the triangle inequality, are the *metric versions* of the problems, for which all the following results are obtained.

In the *facility location problem* (UFL) we are, also, given a non-negative cost f_i of opening a facility at i, for every location $i \in F$. The cost function that has to be minimized for this problem is the sum of the cost of opening the facilities (*facility cost*) and the *shipping (or service) cost*. On the other hand, in the *k-median problem*, instead of facility costs we are just restricted to minimize the service cost, opening at most k facilities ($|S| \leq k$).

More formally, the service cost associated with a set S of open facilities and an assignment of locations to them, $\sigma : N \rightarrow S$, is given by $C_s(S) = \sum_{j \in N} d_j c_{j\sigma(j)}$. The facility cost, for the UFL, is $C_f(S) = \sum_{i \in S} f_i$. For both problems, given a set S of open facilities, an assignment that minimizes the total cost is to assign each location $j \in N$ to the closest open facility in S. Thus, a solution to these problems is completely characterized by the set of open facilities S.

The previous problems are called *uncapacitated* (that is the "U" in "UFL") in the sense that the demand that can be shipped from any facility is infinite. The *capacitated* variants of the above two problems are divided in two categories depending on how the locations' demand can be served. If the demand of each location can be split across more than one facility then we have a *splittable capacitated* variant. If the demand of each location has to be shipped from a single facility then we have an *unsplittable capacitated* variant. Furthermore, the capacities that the facilities have can be *uniform* or *non-uniform*, that is either there is a common bound M for the capacities of all the facilities or each facility $j \in F$ has a specific capacity $u_j > 0$, respectively. For the case of capacitated problems with splittable demands, the assignment function changes and is given by $\sigma : N \times S \rightarrow \mathbb{R}$, where $\sigma(i, j)$ denotes the amount of demand shipped to location i from facility j.

As we have already mentioned, in the uncapacitated problems, given a set of open facilities, an optimal assignment is obtained by simply assigning each location to its closest open facility. In the capacitated variations such an assignment may violate the capacity constraint(s). Fortunately, for the splittable capacitated problems we can compute an optimal assignment in polynomial time, solving an appropriately defined instance of the transportation problem, [44]. However, when the demands are unsplittable, it is NP-hard to compute an optimal assignment for a given set S of open facilities. Therefore, we require a solution to one of the capacitated problems with unsplittable demands to specify a feasible assignment together with the set of open facilities.

Other variations of the capacitated problems are those, which permit multiple *copies* of a facility to be opened in a location. Hence, in that case we are looking for a multi-set S of open facilities. The difference with the uncapacitated problems is that we are only permitted to open at most m copies. Thus, the capacitated facility location problem with at most m copies of each facility permitted, is called *m-CFLP*. The notion of copies is equivalent with that of *capacity blowup*, considered in [45], which is used in capacitated k-median problems.

To sum up, the problems that we have defined are the metric versions of the following problems:

1. uncapacitated k-median and facility location problems (UFL),
2. capacitated k-median and facility location problems with unsplittable demands having uniform or non-uniform capacities, with copies (m-CFLP) or not,
3. capacitated k-median and facility location problems with splittable demands having uniform or non-uniform capacities, with copies (m-CFLP) or not.

In Table 1 we give some approximation ratios for some of these metric (except from the third one which is general) problems, a bound of copies (or capacity blowup) if they are permitted and the reference. Note that for the k-median problems an (a, b)-*approximation algorithm* is defined as a polynomial time algorithm that computes a solution using at most bk facilities and with cost at most a times the cost of an optimal solution using at most k facilities.

We should also refer a theorem proved in [53]. It says that there is a polynomial algorithm which, given a solution S to the k-CFLP, produces a solution $\hat{S}$ to the 2-CFLP at additional cost at most twice the optimal value of a solution to the 1-CFLP.

The results, on Table 1, with an asterisk next to their reference, are obtained with local search and some of them are the best known. Local search seems to work very well with these problems. We will give, now, such an algorithm with its ε-approximability proof. It is the $5(1+\varepsilon)$-approximation algorithm for the uncapacitated k-median problem, presented in [46], whose extension gives a $(3 + 2/p)(1 + \varepsilon)$-approximation algorithm.

Problem	Bound	Copies (Capacity Blowup)	Reference
1. uncapacitated k-median	$(1+\varepsilon, 3+5/\varepsilon)$	-	[45]*
	$(1+5/\varepsilon, 3+\varepsilon)$	-	[45]*
	$5(1+\varepsilon)$	-	[46]*
	$(3+2/p)(1+\varepsilon)$	-	[46]*
	$O(logkloglogk)$	-	[47]
	$(1+\varepsilon, (1+1/\varepsilon)(lnn+1))$	-	[48]
	$(2(1+\varepsilon), 1+1/\varepsilon)$	-	[49]
2. Euclidean k-median	$(1+\varepsilon, 1)$	-	[50]
3. general uncapacitated facility location	$O(logn)$	-	[51]
4. uncapacitated facility location	$3(1+\varepsilon)$	-	[46]*
	1.74	-	[52]
	$\nexists 1.46$, unless $P=NP$	-	[53]
5. k-median, splittable, uniform	$(1+\varepsilon, 5+5/\varepsilon)$	none	[45]*
	$(1+5/\varepsilon, 5+\varepsilon)$	none	[45]*
6. k-median, unsplittable, uniform	$(1+\varepsilon, 5+5/\varepsilon)$	2	[45]*
	$(1+5/\varepsilon, 5+\varepsilon)$	2	[45]*
7. facility location, splittable, uniform	$8+\varepsilon$	none	[45]*
	$6(1+\varepsilon)$	none	[53]*
	7	7/2	[44]
	3	∞	[54]
	5	2	[53]
8. facility location, splittable, non-uniform	$4(1+\varepsilon)$	∞	[46]*
	$9+\varepsilon$	none	[55]*
9. facility location, unsplittable, uniform	$16+\varepsilon$	2	[45]*
	9	4	[44]

Table 1: Approximation bounds for k-median and facility location problems (references with asterisk indicate that the local search method was used). For the uncapacitated problems the notion of copies has no meaning so we put a '-'.

The notation in [46] is different, but equivalent with the one presented here, so we will redefine the problem.

7.6.1 Uncapacitated k-Median Problem

In the metric uncapacitated k-median problem, we are given two sets, F (facilities) and C (clients), and an input parameter $k, 0 < k \leq |F|$. There is a specified *metric distance* $c_{ij} \geq 0$ between every pair $i, j \in F \cup C$, which is used as *service cost*, too. The problem is to identify a subset $S \subseteq F$ of at most k facilities and to serve the clients in C by the facilities in S such that the total service cost is minimized. Thus, if a client $j \in C$ is served by (its closest) facility $\sigma(j) \in S$, then we want to minimize $cost(S) = \sum_{j \in C} c_{\sigma(j)j}$.

The general local search algorithm used in [46] is the following:

```
1. $S \leftarrow$ an arbitrary feasible solution.
2. While $\exists$ an operation $op$ such that,
           $cost(op(S)) \leq (1 - \frac{\epsilon}{p(n,m)})cost(S)$,
     do $S \leftarrow op(S)$.
3. return S.
```

where $n = |F|, m = |C|$ and $p(n, m)$ a polynomial in n and m. The neighborhood used in this local search procedure is *swap*. A swap is effected by closing a facility $s \in S$ and opening a facility $s' \notin S$. So

$$op(S) := S - s + s', \text{ for } s \in S \text{ and } s' \notin S.$$

and this swap will be denoted by $\langle s, s' \rangle$. If the second step's inequality holds, then the operation op is called *admissible* for S. This algorithm terminates in polynomial time, since each swap is performed in polynomial time, the number of swaps being performed is

$$\frac{log(cost(S_0)/cost(S^*))}{log\frac{1}{1-\epsilon/p(n,m))}},$$

where S_0 and S^* are the initial and optimum solutions respectively, and $log(cost(S_0))$ is polynomial in the input size.

When there are no admissible operations then we know that every operation reduces the cost by a factor of at most $\varepsilon/p(n, m)$, i.e. $cost(op(S)) \geq (1 - \frac{\epsilon}{p(n,m)})cost(S)$. To simplify the exposition, the assumption $cost(op(S)) \geq cost(S)$ is used. So, by adding at most $p(n, m)$ of such inequalities we can conclude that $cost(S) \leq \alpha \cdot cost(S^*)$ for some $\alpha \geq 1$, that is a *locality gap* α. Adding the corresponding original inequalities implies that $cost(S) \leq \alpha(1 + \epsilon)cost(S^*)$, that is an $\alpha(1 + \epsilon)$-approximation.

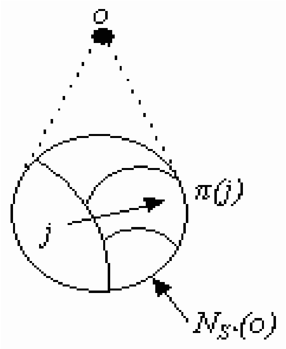


Figure 3: A matching π on $N_{S^*}(o)$

The following notation is used. Let s_j and o_j denote the service costs of a client j in the solutions S and S^* respectively. Let $N_S(s)$ denote the set of clients in C that are served by a facility $s \in S$ in the solution S. Similarly $N_{S^*}(o)$ denotes the set of clients in C that are served by a facility $o \in S^*$ in the solution S^*. Finally, for a subset $A \subseteq S$, let $N_S(A) = \cup_{s \in A} N_S(s)$.

Now we are ready to show that the local search procedure as defined above has a locality gap of 5. From the local optimality of S, we know that any swap $\langle s, o \rangle$ for $s \in S$ and $o \in S^*$,

$$cost(S - s + o) \geq cost(S) \text{ for all } s \in S, o \in S^*. \tag{4}$$

Combining these inequalities we can show that $cost(S) \leq 5 \cdot cost(S^*)$. Note that the algorithm and its analysis extend simply to the case when the clients $j \in C$ have arbitrary demands $d_{ij} \geq 0$ to be served. Also, the extension of this neighborhood to a p-Opt, where up to p facilities can be swapped simultaneously, has a $3 + 2/p$ locality gap, which is tight (see [46]). So, we have

Theorem 7.8 *A local search procedure for the metric k-median problem with operations defined as $op(S) := S - s + s'$ for $s \in S$ and $s' \notin S$, has a locality gap at most 5.*

Proof. Consider a facility $o \in S^*$. We partition $N_{S^*}(o)$ into subsets $p_s = N_{S^*}(o) \cap N_S(s)$ for $s \in S$. Consider a 1-1 and onto mapping $\pi : N_{S^*}(o) \to N_{S^*}(o)$ with the following property: for all $s \in S$ such that, $|p_s| \leq \frac{1}{2}|N_{S^*}(o)|$, we have, $\pi(p_s) \cap p_s = \varnothing$. It is easy to see that such a mapping π exists, (Fig. 3).

We say that a facility $o \in S^*$ is *captured* by a facility $s \in S$ if s serves more than half of the clients served by o, that is, $|N_S(s) \cap N_{S^*}(o)| > \frac{1}{2}|N_{S^*}(o)|$. Note that a facility $o \in S^*$ is captured by at most one $s \in S$. We call facility $s \in S$ *bad* if it captures some facility in S^* and *good* otherwise.

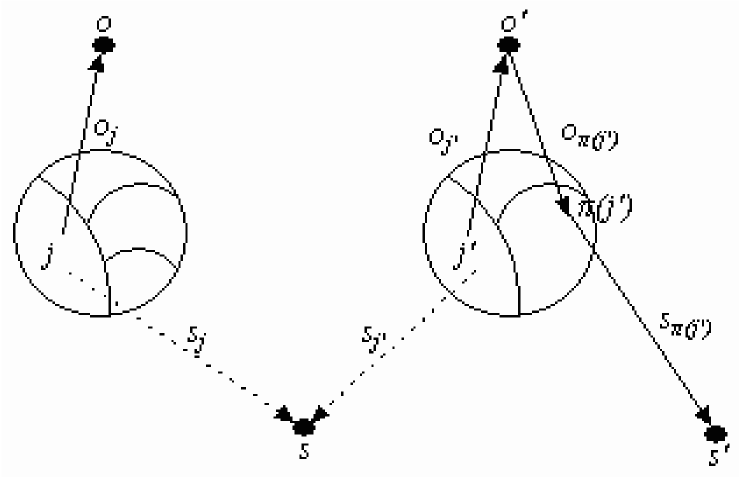


Figure 4: Reassigning the clients in $N_S(s) \cup N_{S^*}(o)$

We now consider k swaps, one for each facility in S^*. If some bad facility $s \in S$ captures exactly one facility $o \in S^*$ then we consider the swap $\langle s, o\rangle$. Suppose l facilities in S (and hence l facilities in S^*) are not considered in such swaps. These l facilities in S are either good or bad, and the bad facilities capture at least two facilities in S^*. Hence, there are at least $l/2$ good facilities in S. Now, consider l swaps in which the remaining l facilities in S^* get swapped with the good facilities in S such that each good facility is swapped-out at most twice.

It is easy to verify that the swaps considered above satisfy the following properties:

1. Each $o \in S^*$ is swapped-in exactly once.

2. Each $s \in S$ is swapped out at most twice. This is because a facility in S that captures more than one facility in S^* is never swapped-out and a facility that capture exactly one facility in S^* is swapped only with the facility that it captures.

3. If a swap $\langle s, o\rangle$ is considered, the facility s does not capture any facility $o' \neq o$.

We now analyze these swaps by considering an arbitrary swap $\langle s, o\rangle$. We place an upper bound on the increase in cost due to this swap by reassigning the clients in $N_S(s) \cup N_{S^*}(o)$ to the facilities in $S - s + o$ as follows (see Fig. 4). The clients $j \in N_{S^*}(o)$ are now assigned to o. Consider a client $j' \in N_S(s) \cap N_{S^*}(o')$, for $o' \neq o$. As s does not capture o', we have $|N_S(s) \cap N_{S^*}(o')| \leq \frac{1}{2}|N_{S^*}(o')|$ and hence by the property of π, we have that $\pi(j') \notin N_S(s)$. Let $\pi(j') \in N_S(s')$. Note that the distance the client j' travels to the nearest facility in $S - s + o$ is at most $c_{j's'}$. Also from triangle inequality,

$c_{j's'} \leq c_{j'o} + c_{o\pi(j')} + c_{\pi(j')s'} = o_{j'} + o_{\pi(j')} + s_{\pi(j')}$. The remaining clients continue to be assigned to the old facilities. From inequality 4 we have,

$$cost(S - s + o) - cost(S) \geq 0.$$

Therefore,

$$\sum_{j \in N_{S^*}(o)} (o_j - s_j) + \sum_{j \in N_S(s), j \notin N_{S^*}(o)} (o_j + o_{\pi(j)} + s_{\pi(j)} - s_j) \geq 0 \qquad (5)$$

As each facility $o \in S^*$ is swapped-in exactly once, the first term of the inequality 5 added over all the k swaps gives exactly $cost(S^*) - cost(S)$. For the second term, we use the fact that each s is swapped-out at most twice. Also for any $j \in C$, as s_j is the shortest distance from j to a facility in S, we get, using triangle inequality, $o_j + o_{\pi(j)} + s_{\pi(j)} \geq s_j$. Thus the second term of the inequality 5 added over all the k swaps is not greater than $2\sum_{j \in C}(o_j + o_{\pi(j)} + s_{\pi(j)} - s_j)$. But as π is 1-1 and onto mapping, $\sum_{j \in C} o_j = \sum_{j \in C} o_{\pi(j)} = cost(S^*)$ and $\sum_{j \in C}(s_{\pi(j)} - s_j) = 0$. Thus, $2\sum_{j \in C}(o_j + o_{\pi(j)} + s_{\pi(j)} - s_j) = 4 \cdot cost(S^*)$. Combining the two terms we get $cost(S^*) - cost(S) + 4 \cdot cost(S^*) \geq 0$. □

7.6.2 Capacity Allocation Problem

The capacity allocation problem (CAP) is a multi-commodity generalization of the single-commodity k-median problem, involving multiple types of service and the requirement that all nodes receive all these types from the corresponding supply nodes. This problem has applications in Internet content distribution. In [56], an exact algorithm that solves the problem, solving first a sufficient number of k-median problems, is presented. The combination of this algorithm with a polynomial time constant factor approximation algorithm for the k-median problem yields an approximation ratio for CAP as good as the one for the k-median. The extension of the algorithm, that we described above, to swaps of up to p facilities simultaneously, is the best known and has a $(3 + 2/p)(1 + \varepsilon)$-approximation ratio. Thus the CAP problem has a polynomial time $(3 + 2/p)(1 + \varepsilon)$-approximate algorithm.

7.6.3 k-means Clustering

In k-means clustering we are given a set of n data points in d-dimensional space $\mathcal{R}^d$ and an integer k, and the problem is to determine a set of k points in $\mathcal{R}^d$, called centers, to minimize the mean squared distance from each data point to its nearest center. For this problem no exact polynomial-time algorithm is known and although, asymptotically efficient algorithms exist (see [57]), they are not practical.

The main difference of this problem and the metric k-median, and thus the main difficulty of applying ones results on the other, is that in the first case the triangle inequality does not hold (however the doubled triangle inequality holds).

An iterative heuristic, called Lloyd's algorithm (see [58]), exists but it can converge to a local minimum arbitrarily worst than the global one. It starts with any feasible solution and then repeatedly computes the "neighborhood" of each center (the data points closest to it) and moves this center to the centroid of its "neighborhood".

In [59], a $(9 + \varepsilon)$-approximation local search algorithm is presented, based on the previous algorithm for k-median in [46], that we presented. It is based on swapping centers in and out of the solution set. This algorithm combined with the previous one has empirically shown a good practical performance.

7.7 Quadratic Assignment Problem

Given two $n \times n$ symmetric matrices $F = (f_{ij})$ and $D = (d_{ij})$, with a null diagonal, the symmetric Quadratic Assignment Problem (QAP) can be stated as follows:

$$\min_{\pi \in \Pi} \sum_{i=1}^{n} \sum_{k=i+1}^{n} f_{ik} d_{\pi(i)\pi(k)},$$

where Π is the set of all permutations of $\{1, 2, \ldots, n\}$. One of the major applications of the QAP is in location theory where f_{ij} is the flow of materials from facility i to facility j, and d_{ij} represents the distance from location i to location j. The objective is to find an assignment of all facilities to locations which minimizes the total cost.

The 2-exchange neighborhood is usually applied on this problem. That is, given a permutation $\pi = (\pi(1), \ldots, \pi(i), \ldots, \pi(j), \ldots, \pi(n))$, its neighbors are the $\frac{n(n-1)}{2}$ permutations of the form $\pi = (\pi(1), \ldots, \pi(j), \ldots, \pi(i), \ldots,$
$\pi(n))$ for $1 \leq i \leq j \leq n$, obtained from π by a swap.

QAP is NP-hard. Since Graph Partitioning under the swap neighborhood is a special case of the symmetric QAP under the 2-exchange neighborhood and as we have already seen it is PLS-complete, it follows that QAP under 2-exchange is PLS-complete, too.

In [60], a result has been obtained for QAP, which also applies to symmetric Travelling Salesman Problem, Graph Partitioning, k-Densest Subgraph, k-Lightest Subgraph and Maximum Independent Set, as they are subcases of QAP.

At first we give some notation. Let $s(A)$ denote the sum of all terms of a given matrix A. Let x and y two vectors of the same dimension. The maximum (resp. minimum) scalar product of x and y is defined by: $\langle x, y \rangle_+ = max_{\pi \in \Pi} \langle x, \pi y \rangle$ (resp. $\langle x, y \rangle_- = min_{\pi \in \Pi} \langle x, \pi y \rangle$). Let F_k and D_k denote the sum over the kth column of F and D, respectively. Let $\langle F, D \rangle_+$ (resp. $\langle F, D \rangle_-$) be an abbreviation for $\langle (F_1, \ldots, F_n), (D_1, \ldots, D_n) \rangle_+$

(resp. $\langle (F_1, \ldots, F_n), (D_1, \ldots, D_n) \rangle_-$). The following two theorems and the corollary have been proved:

Theorem 7.9 *For the QAP, let C^-_{loc} the cost of any solution found by a deepest descent local search*[4] *with the 2-exchange neighborhood, then the following inequality holds:*

$$C^-_{loc} \leq \frac{\langle F, D \rangle_-}{s(F)s(D)} nC_{AV},$$

where C_{AV} is the average cost of all possible permutations.

Corollary 7.1 *For the QAP, the following inequality holds:*

$$C^-_{loc} \leq \frac{n}{2} C_{AV}.$$

Moreover, there is a sequence of instances for which the ratio $C_{max}/\frac{n}{2}C_{AV}$ tends to infinity, where C_{max} is the maximum cost over all permutations.

Theorem 7.10 *If the matrices F and D are positive integer ones, a deepest descent local search will reach a solution with a cost less than $\frac{n}{2}C_{AV}$ in at most $O(nlog(\frac{s(F)s(D)}{2})$ iterations.*

Notice that, when one of the matrices, say F, has constant row sums, i.e. $Fe = \lambda e$, for e a vector of all ones, then $\langle F, D \rangle_+ / s(F)s(D) = 1/n$ and it follows that $C^-_{loc} \leq C_{AV}$ from theorem 7.9. Now, let us see the applications of the above theorems on some known problems.

7.7.1 The Symmetric Travelling Salesman Problem

This problem can be seen as a particular case of QAP by considering D to be the distance matrix and F to be defined by $f_{i,i+1} = f_{i+1,i} = 1$ with $1 \leq i \leq n-1, f_{n,1} = f_{1,n} = 1$ and $f_{ij} = 0$ otherwise. Using the above remark, it is obtained, $C^-_{loc} \leq C_{AV}$, for the 2-exchange neighborhood, which is the same as 2-Opt.

[4] That is, the local search heuristic which successively replaces the current solution by the *best* neighboring one.

7.7.2 The unweighted Graph Partitioning Problem

Recall that in this problem we are given a graph and we have to partition its vertices in two equal-sized subsets A and B, such that the number of edges having one extremity in A and the other in B, is minimized. For this problem, D is the adjacency matrix of the graph, and

$$F = \begin{pmatrix} 0 & U \\ U & 0 \end{pmatrix}$$

where U is the $n/2 \times n/2$ matrix, with $u_{ij} = 1, i, j = 1, \ldots, n/2$. Using the above remark, it is obtained for the swap neighborhood, $C^{-}_{loc} \leq C_{AV}$.

7.7.3 The unweighted k-Lightest and the k-Densest Subgraph Problems

These problems are defined as follows. Given a graph $G = (V, E)$ and a number $m(m \leq |V|)$, find m vertices of G such that the number of edges in the subgraph induced by these vertices is minimum (respectively maximum). These problems have also been studied in [60], but they were referred to as Generalized Maximum independent Set and Generalized Maximum Clique Problems. They can be modelized by a QAP with D the adjacency matrix of graph G and $F = (f_{ij})$, where $f_{ij} = 1$ if $i \neq j, 1 \leq i, j, \leq m$ and $f_{ij} = 0$ otherwise. In the sequel it is considered that $d_1, d_2, \ldots, d_n$ are the degrees of the vertices of G arranged in decreasing order. The following result was obtained.

Proposition 7.1 *The local optimal solution found by a deepest local search with the swap neighborhood satisfies $C^{-}_{loc} \leq ((m-1)/2(n-1))(d_1 + d_2 + \ldots + d_m)$ (respectively $C^{+}_{loc} \geq ((m-1)/2(n-1))(d_n + d_{n-1} + \ldots + d_{n-m+1})$) for the minimization (respectively the maximization problem).*

7.7.4 The Maximum Independent Set Problem

Finally for MIS we have the following proposition

Proposition 7.2 *If $d_1 + d_2 + \ldots + d_k \leq \lfloor 2c(n-1)/(k-1) \rfloor_*$, with $2 \leq k \leq n$ and c any integer, the deepest local search with the swap neighborhood finds an independent set with at least $k - c + 1$ vertices. By definition, $\lfloor x \rfloor_*$ is equal to $x - 1$ if x is an integer, and $\lfloor x \rfloor$ otherwise.*

8 Conclusions and open problems

Local Search is a method extensively used to approximately solve NP-hard Combinatorial Optimization Problems. The aim of this work was to sum up some main theoretical results that we have for Local Search. So, at first, we saw the theory of PLS-completeness, which gives us the instruments to recognize the difficulty of Local Search Problems and we concluded that for the PLS-complete problems the standard local search heuristic takes exponential time in the worst case.

Then, we saw that the quality of the local optima depends on the NP-hardness of the corresponding optimization problems. Theorem 6.1 provides negative indications for the approximation efficiency of local search on NP-hard optimization problems. However, apart from the experimentally observed power of local search and the probabilistic verification of this ability, there are lately, a lot of results, which provide ε-approximation guarantees of local search for many common NP-hard problems. Section 7 presents these results giving also some characteristic proofs. All the problems mentioned there, are either unweighted or with polynomially bounded weights subcases of their general problems. Since local search is pseudopolynomial, the standard local search heuristic terminates in a rather competitive polynomial time for them.

There are two more theoretical questions concerning local search. The first one is about the parallel complexity of determining if we are on a local optimum solution and computing a better neighbor if we are not. Generally this problem is independent of the difficulty of the local search problem itself. Hence, for Max-Sat/Flip the complexity is in NC while Graph Partitioning/KL is P-complete (see [4]), both problems being PLS-complete. The second question asks about the complexity of the standard local optimum problem. That is, for a given problem, starting from a specific solution, how fast we can find the local optimum that the standard local search heuristic would have produced. It turns out, that for all PLS-complete problems this latter problem is PSPACE-complete (see [4]).

Closing this work, it would be a great lack not to refer to some other uses of local search. First of all, due to its efficiency and simplicity, local search methods are also used to solve polynomial problems, such as Linear Programming, Maximum Matching and Maximum Flow. The well-known algorithm Simplex, is a local search heuristic, which explores an exact neighborhood (each time, the adjacent vertices of the current vertice on the polytope) and it is proved to take exponential time in the worst case for many pivoting rules. Simplex is used to solve Continuous Linear Programming Problems, despite the existence of polynomial algorithms, such as the interior point algorithms, because it works much better in practice.

Another use of local search is as a tool in proofs of existence of a solution to a problem, i.e. that a search problem is total. For example, we saw in subsection 3.6 that the proof, that there is always a stable configuration in neural networks of the Hopfield model, depends on proving that the initial search problem can be transformed into a local search problem, under appropriate cost and neighborhood functions, and hence the existence of the stable configurations are guaranteed by the existence of the local optima. In [4], there is the Submatrix problem proposed by Knuth, for which the same argument guarantees the existence of its solution.

Recently, in [61], a connection between PLS Theory and Game Theory established. More particularly, the problems of finding *pure Nash equilibria* in General Congestion Games, Symmetric Congestion Games and Asymmetric Network Congestion Games were shown to be PLS-complete. The reductions follow from the Pos NAE-3Sat/Flip. The use of local search in proofs of existence, referred in the previous paragraph, is also applied to proofs of existence of pure Nash equilibria in games, called as the potential function method. Let's call a game as a *general potential function game* if there is a function ϕ such that for any edge of the Nash dynamics graph (s, s') with defector i we have $sgn(\phi(s') - \phi(s)) = sgn(u_i(s') - u_i(s))$. These games, obviously have pure Nash equilibria, from the potential function argument. An interesting result, presented in [61], is a converse one, that the class of general potential games essentially comprises all of PLS.

An extension of local search, proposed in [18] as a general paradigm useful for developing simple yet efficient approximation algorithms, is *non-oblivious local search.* Non-oblivious local search allows the cost function, used to find a local optimum, to be different from that of the original problem, hence the search can be directed to better quality solutions. It is showed that every MAX-SNP problem can be approximated to within constant factors by this method. Ausiello and Protasi defined, in [62], the class GLO (guaranteed local optima) of combinatorial optimization problems which have the property that for all locally optimum solutions, the ratio between the value of the global and the local optimum is bounded by a constant. Vertex Covering does not belong to GLO but it is MAX-SNP, hence GLO is a strict subset of non-oblivious GLO.

Furthermore, local search is the base of most *metaheuristics*. In [63], there is an overview on metaheuristics and the following definition of Stützle, [64], is given among others: "Metaheuristics are typically high-level strategies which guide an underlying, more problem specific heuristic, to increase their performance. The main goal is to avoid the disadvantages of iterative improvement and, in particular, multiple descent by allowing the local search to escape from local optima. This is achieved by either allowing worsening moves or generating new starting solutions for the local search in a more "intelligent" way than just providing random initial solutions". Some of the most common metaheuristics are Tabu Search, Simulated Annealing, GRASP, Iterated Local Search, Variable Neighborhood Search, Evolutionary Computation and Ant Colony Optimization. Apart from the last one, all the other methods are variations of local search, in which the cost function or the neighborhood or the allowed perturbations on a solution change dynamically during the search. The performance of such methods is studied empirically and it would be rather difficult to have some purely theoretical results about them.

Finally, there is the *landscape theory*, which tries to theoretically justify why a neighborhood is better in practice than another for a given optimization problem. The ruggedness of the landscape which is formed by the cost function and the neighborhood is measured, since a good agreement between ruggedness and difficulty for local search is observed. A hierarchy of the combinatorial optimization problems can be obtained relatively to their ruggedness. More about this field one can find in [65, 66].

There are a lot of theoretical and experimental *open problems* in the area of local search. First of all, we do not know the exact relation of the class PLS with the classes P and NP. There are, also, many interesting local search problems, such as TSP/2-Opt, that we do not know if they are PLS-complete or not. Furthermore, a lot of NP-hard problems have approximately been solved, within constant factors, by local search methods, so improvement of these factors and extension of such results to more problems, would be of great importance. On the experimental aspect of research of this field, the average performance of local search algorithms both in computational time and in approximation efficiency, is rather interesting. In landscape theory there are unanswered questions about the definition of the ruggedness and other characteristics of the landscapes of local search problems. Neighborhoods of exponential size are under research too, in order to examine whether we can search them efficiently or if we can guarantee ε-local optima with them. Finally, the recent use of local search methods in game theory seems to give some first, very interesting results.

References

[1] D.S. Johnson, C.H. Papadimitriou & M. Yannakakis. *How easy is local search?* Journal of Computer and System Sciences (1988), **37(1)**, pp. 79-100.

[2] B.W. Kernighan & S. Lin. *An efficient heuristic procedure for partioning graphs.* Bell System Technical Journal (1970), **49**, pp. 291-307.

[3] P. Christopoulos. *Local Search and PLS-completeness.* Undergraduate Thesis, Department of Informatics and Telecommunications, University of Athens (2003).

[4] M. Yannakakis. *Computational Complexity.* In E. Aarts and J. k. Lenstra, *Local Search in Combinatorial Optimization,* Wiley-Interscience Publication (1998), Chapter 2, pp. 19-55.

[5] C.H. Papadimitriou. *The complexity of the Lin-Kernighan heuristic for the TSP.* SIAM Journal on Computing (1992), **21**, pp. 450-465.

[6] J.J. Hopfield. *Neural networks and physical systems with emergent collective computational abilities.* Proceedings of the National Academy of Sciences of the USA (1982), **79**, pp. 2554-2558.

[7] J. Bruck & J.W. Goodman, *A generalized convergence theorem for neural networks*, IEEE Transactions on Information Theory (1988) **34**, pp. 1089-1092.

[8] G. Godbeer, *On the computational complexity of the stable configuration problem for the connectionist models*, MSc thesis (1987), Department of Computer Science, University of Toronto.

[9] J. Lipscomb, *On the computational complexity of finding a connectionist model's stable state of vectors*, MSc thesis (1987), Department of Computer Science, University of Toronto.

[10] A.A. Schaffer & M. Yannakakis. *Simple local search problems that are hard to solve.* SIAM Journal on Computing (1991), **20(1)**, pp. 56-87.

[11] M.W. Krentel. *Structure in locally optimal solutions.* 30th Annual Symposium on Foundations of Computer Science (1989), IEEE Computer Society Press, Los Alamitos, CA, pp. 216-222.

[12] M.W. Krentel. *On finding and verifying locally optimal solutions.* SIAM J. on Computing (1990), **19**, pp. 742-751.

[13] J.B. Orlin, A.P. Punnen, A.S. Schulz. *Approximate Local Search In Combinatorial Optimization.* (July 2003), MIT Sloan Working Paper No. 4325-03.

[14] N. Christofides. *Worst-case analysis of a new heuristic for the travelling salesman problem.* Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA, (1976).

[15] C.H. Papadimitriou. *On selecting a satisfying truth assignment.* In Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science (1991), FOCS'91, pp. 163-169.

[16] E. Dantsin, A. Goerdt, E.A. Hirsch, R. Kannan, J. Kleinberg, C. Papadimitriou, P. Raghavan and U. Schöning. *A deterministic $(2 - \frac{2}{k+1})^n$ algorithm for k-SAT based on local search.* (2001), http://citeseer.nj.nec.com/dantsin01deterministic.html.

[17] P. Hansen and B. Jaumard. *Algorithms for the maximum satisfiability problem.* Computing (1990), **44**, pp. 279-303.

[18] S. Khanna, R. Motwani, M. Sudan, U. Vazirani. *On syntactic versus computational views of approximability.* Technical Report TR95-023, Ellectronic colloquium on computational complexity (1995), http://www.eccc.uni-trier.de/eccc/.

[19] R. Battiti and M. Protasi. *Solving MAX-SAT with non-oblivious functions and history-based heuristics.* In *Satisfiability problems: Theory and applications*, DIMACS: Series in discrete mathematics and theoretical computer science (1997), no. 35, AMS and ACM Press.

[20] J. Håstad. *Some optimal inapproximability results.* Proceedings of the 29th ACM Symposium on the Theory of Computation (1997), ed. L. Longpré, ACM, New York, pp. 1-10.

[21] F. Alizadeh. *Optimization over the positive semi-definite cone: interior point methods and combinatorial applications.* P.M. Pardalos (Ed.), Advances in Optimization and Parallel Computing, North-Holland, Amsterdam, Netherlands, (1992), pp. 1-25.

[22] C. Helmberg, F. Rendl, R.J. Vanderbei, H. Wolkowicz. *An interior point method for semidefinite programming.* Technical Report (1994), University of Graz.

[23] G. Ausiello. *Compexity and approximation: Combinatorial optimization problems and their approximation properties.* Springer (1999).

[24] A. Bertoni, P. Campadelli, G. Grossi. *An approximation algorithm for the maximum cut problem and its experimental analysis.* Discrete Applied Mathematics (2001), **110**, pp. 3-12.

[25] C.H. Papadimitriou & K. Steiglitz. *On the complexity of local search for the TSP.* SIAM Journal of Computing (1977), **6**, pp. 76-83.

[26] S. Sahni & T. Gonzales. *P-complete approximation problems.* Journal of the Association for Computing Machinery (1976), **23**, pp. 555-565.

[27] B. Chandra, H. Karloff, C.A. Tovey. *New results on the old k-opt algorithm fot the TSP.* SIAM Journal on Computing (1999), **28(6)**, pp. 1998-2029.

[28] M. Halldórsson. *Approximating discrete collections via local improvements.* Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (1995), ACM New York and SIAM Philadelphia PA, pp. 160-169.

[29] B. Chandra and M.M. Halldórsson. *Greedy local improvement and weighted packing approximation.* In SODA (1999), pp. 169-176.

[30] B. Chandra and M.M. Halldórsson. *Greedy local improvement and weighted packing approximation.* Journal of Algorithms (2001), **39(2)**, pp. 223-240.

[31] P. Berman. *A $d/2$ approximation for maximum weight independent set in d-claw free graphs.* Nordic Journal of Computing (2000), **7(3)**, pp. 178-184.

[32] R. Duh and M. Fürer. *Approximation of k-set cover by semi-local optimization.* ACM Symposium on Theory of Computing (1997), pp. 256-264.

[33] S. Khuller, R. Bhatia, R. Pless. *On local search and placement of meters in networks.* Symposium on Discrete Algorithms (2000), pp. 319-328.

[34] J. Kleinberg, É. Tardos. *Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields.* In Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science (1999), pp. 14-23.

[35] J. Besag. *On the statistical analysis of dirty pictures.* J. Royal Statistical Society B (1986), **48(3)**, pp. 259-302.

[36] D. Greig, B.T. Porteous, A. Seheult. *Exact maximum a posteriori estimation for binary images.* J. Royal Statistical Society B (1989), **51(2)**, pp. 271-279.

[37] Y. Boyjov, O. Veksler, R. Zabih. *Markov random fields with efficient approximations.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE computer Society Press (1998), Los Alamitos, Calif, pp. 648-655.

[38] O. Veksler. *Efficient graph-based energy minimization methods in computer vision.* PhD Thesis (1999), Department of Computer Science, Cornell University.

[39] H. Ishikawa, D. Geiger. *Segmentation by grouping junctions.* In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (1999), pp. 125-131.

[40] A. Karzanov. *Minimum 0-extension of graph metrics.* Europ. J. Combinat. (1998), **19**, pp. 71-101.

[41] A. Karzanov. *A combinatorial algorithm for the minimum $(2,r)$-metric problem and some generilizations.* Combinatorica (1999), **18(4)**, pp. 549-569.

[42] C. Chekuri, S. Khanna, J. Naor, L. Zosin. *Approximation algorithms for the metric labeling problem via a new linear programming formulation.* Symposium on Discrete Algorithms (2001), pp. 109-118.

[43] A. Gupta, É. Tardos. *A constant factor approximation algorithm for a class of classification problems.* In Proceedings of the 32nd Annual ACM Symposium on the Theory of Computating (2000), pp. 652-658

[44] D.B. Shmoys, É. Tardos, K. Aardal. *Approximation algorithms for facility location problems.* In Proceedings of the 29th Annual ACM Symposium on Theory of Computing (1997), pp. 265-274.

[45] M. Korupolu, C. Plaxton, R. Rajaraman. *Analysis of a local search heuristic for facility location problems.* Technical Report 98-30, DIMACS, June 1998.

[46] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Mungala, V. Pandit. *Local search heuristic for k-median and facility location problems.* Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing (2001), pp. 21-29.

[47] M. Charikar, C. Chekuri, A. Goel, S. Guha. *Rounding via trees: Deterministic approximation algorithms for group steiner trees and k-median.* In Proceedings of the 30th Annual ACM Symposium on Theory of Computing (1998), pp. 106-113.

[48] J.H. Lin, J.S. Vitter. *ε-approximations with minimum packing constraint violation.* In Proceedings of the 24th Annual ACM Symposium on Theory of Computing (1992), pp. 771-782.

[49] J.H. Lin, J.S. Vitter. *Approximation algorithms for geometric median problems.* Information Processing Letters (1992), **44**, pp. 245-249.

[50] S. Arora, P. Raghavan, S. Rao. *Approximation schemes for Euclidean k-medians and related problems.* In Proceedings of the 30th Annual ACM Symposium on Theory of Computing (1998), pp. 106-113.

[51] D.S. Hochbaum. *Heuristics for the fixed cost median problem.* Mathematical Programming (1982), **22**, pp. 148-162.

[52] F.A. Chudak. *Improved approximation algorithms for the uncapacitated facility location problem.* In Proceedings of the 6th Conference on Integer Programming and Combinatorial Optimization (1998). pp. 180-194.

[53] F.A. Chudak and D.P. Williamson. *Improved approximation algorithms for capacitated facility location problems.* Proceedings of the 7th International IPCO Conference (1999).

[54] F. Chudak and D.B. Schmoys. *Improved approximation algorithms for a capacitated facility location problem.* In Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (1999), pp. 875-876

[55] M. Pál, É. Tardos and T. Wexler. *Facility location with nonuniform hard capacities.* IEEE Symposium on Foundations of Computer Science (2001), pp. 329-338.

[56] N. Laoutaris, V. Zissimopoulos, I. Stavrakakis. *Joint object placement and node dimensioning for Internet content distribution.* Information Processing Letters, to appear.

[57] J. Matoušek. *On approximate geometric k-clustering.* Discrete and Computational Geometry (2000), **24**, pp. 61-84.

[58] Q. Du, V. Faber, M. Gunzburger. *Centroidal Voronoi tesselations: Applications and algorithms.* SIAM review (1999), **41**, pp. 637-676.

[59] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu. *A local search approximation algorithm for k-means clustering.* In Proceedings of the 18th Annual Symposium on Computational Geometry (2002), Barcelona, Spain, pp. 10-18.

[60] E. Angel, V. Zissimopoulos. *On the quality of local search for the quadratic assignment problem.* Discrete Applied Mathematics (1998), **82**, pp. 15-25.

[61] A. Fabrikant, C. Papadimitriou, K. Talwar. *The complexity of pure Nash equilibria.* Papadimitriou's home page (2003), http://www.cs.berkeley.edu/ christos.

[62] G. Ausiello, M. Protasi. *Local search, reducibility and approximability of NP optimization problems.* Inform. Process. Lett. (1995), **54**, pp. 73-79.

[63] C. Blum, A. Roli. *Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison.* ACM Computing Surveys (2003), **35:3**, pp. 268-308.

[64] T. Stützle. *Local search algorithms for combinatorial problems - analysis, algorithms and new applications.* DISKI - Dissertationen zur Künstliken Intelligenz. infix, Sankt Augustin, Germany.

[65] E. Angel, V. Zissimopoulos. *On the classification of NP-complete problems in terms of their correlation coefficient.* Discrete Applied Mathematics (2000), **99**, pp. 261-277.

[66] C.M. Reidys, P.F. Stadler. *Combinatorial Landscapes.* SIAM Review (2002), **44**, pp. 3-54.

An alternative proof of SAT NP-completeness

Bruno Escoffier*, Vangelis Th. Paschos*

Résumé

Nous donnons une preuve de la **NP**-complétude de SAT en se basant sur une caractérisation logique de la classe **NP** donnée par Fagin en 1974. Ensuite, nous illustrons une partie de la preuve en montrant comment deux problèmes bien connus, le problème de MAX STABLE et de 3-COLORATION peuvent s'exprimer sous forme conjonctive normale. Enfin, dans le même esprit, nous redémontrons la **min NPO**-complétude du problème de MIN WSAT sous la stricte-réduction.

Mots-clefs : logique du second ordre, **NP**-complétude, réductions.

Abstract

We give a proof of SAT's **NP**-completeness based upon a syntaxic characterization of **NP** given by Fagin at 1974. Then, we illustrate a part of our proof by giving examples of how two well-known problems, MAX INDEPENDENT SET and 3-COLORING, can be expressed in terms of CNF. Finally, in the same spirit we demonstrate the **min NPO**-completeness of MIN WSAT under strict reductions.

Key words : **NP**-completeness, reductions, second order logic.

1 Proof of Cook's theorem

According to Fagin's characterization for **NP** ([3]), any $\Pi \in \mathbf{NP}$ can be written in the following way. Assume a finite structure $(U, \mathcal{P})$ where U is a set of variables, called the *universe* and $\mathcal{P}$ is a set of predicates $P_1, P_2, \ldots, P_\ell$ of respective arities $k_1, k_2, \ldots, k_\ell$.

* LAMSADE, Université Paris-Dauphine, 75775 Paris cedex 16, France. {escoffier,paschos}@lamsade.dauphine.fr

Pair $(U, \mathcal{P})$ is an instance of Π. Solving Π on this instance consists of determining a set $\mathcal{S} = \{S_1, S_2, \dots S_p\}$ of predicates on U satisfying a logical formula of the form: $\Psi(\mathcal{P}, S_1, S_2, \dots, S_p)$. In other words, an instance of Π consists of the specification of $P_1, P_2, \dots, P_\ell$ and of U; it is a *yes*-one if one can determine a set of predicates $\mathcal{S} = \{S_1, S_2, \dots S_p\}$ satisfying $\Psi(\mathcal{P}, \mathcal{S})$.

As an example, consider 3-COLORING, where one wishes to answer if the vertices of a graph G can be legally colored with three colors. Here, finite structure $(U, \mathcal{P}) = (V, G)$, where $V = \{v_1, \dots, v_n\}$ is the vertex set of G. This graph is represented by predicate G of arity 2 where $G(x, y)$ iff vertex x is adjacent to vertex y. A graph G is 3-colorable iff:

$$\begin{aligned}
\exists S_1 \exists S_2 \exists S_3 \quad & \Big(\forall x S_1(x) \vee S_2(x) \vee S_3(x)\Big) \\
\wedge & \bigg(\forall x \Big(\neg S_1(x) \wedge \neg S_2(x)\Big) \vee \Big(\neg S_1(x) \wedge \neg S_3(x)\Big) \vee \Big(\neg S_2(x) \wedge \neg S_3(x)\Big)\bigg) \\
\wedge & \bigg(\forall x \forall y \Big(\big(S_1(x) \wedge S_1(y)\big) \vee \big(S_2(x) \wedge S_2(y)\big) \vee \big(S_3(x) \wedge S_3(y)\big)\Big) \Rightarrow \neg G(x, y)\bigg)
\end{aligned}$$

The rest of this section is devoted to the proof of the **NP**-completeness of SAT, i.e., to an alternative proof of the seminal Cook's theorem ([2]). In fact, we will prove that any instance of a problem Π in **NP** (expressed as described previously) can be transformed in polynomial time into a CNF (i.e., an instance of SAT) in such a way the latter is satisfiable iff the former admits a model.

Let Π be a problem defined by $\exists \mathcal{S} \Psi(\mathcal{P}, \mathcal{S})$. Without loss of generality, we can rewrite $\Psi(\mathcal{P}, \mathcal{S})$ in prenex form and redefine Π as $\exists \mathcal{S} Q_1(x_1) \dots Q_r(x_r) \Phi(x_1, \dots, x_r, \mathcal{P}, \mathcal{S})$, where Q_i, $i = 1, \dots, r$, are quantifiers and Φ quantifier-free.

In the first part of the proof, we are going to build in polynomial time a formula φ (depending on Π and on its instance represented by $\mathcal{P} = P_1, P_2, \dots, P_\ell$) such that φ is satisfiable iff there exists $\mathcal{S}$ satisfying formula $\Psi(\mathcal{P}, \mathcal{S})$ (recall that $\mathcal{S}$ is a p-tuple of predicates $S_1, S_2, \dots, S_p$). Then, we will show how one can modify construction above in order to get a CNF φ_S (instance of SAT) satisfiable iff φ do so.

We first build φ. For this, denote by r_i the arity of predicate S_i in the second-order formula describing Π, and by r the number of its quantifiers. Note that neither r_i's nor r depend on the instance of Π (the dependence of Φ on this instance is realized via predicates $P_i(x_{i_1}, \dots, x_{i_{k_i}})$).

Consider an instance of Π, and denote by $v_1, v_2, \dots, v_n$ the variables of set U. We will build a formula φ on $\sum_{j=1}^{p} n^{r_j}$ variables $y^j_{i_1, i_2, \dots, i_{r_j}}$, where $j \in \{1, \dots, p\}$ and $(i_1, i_2, \dots, i_{r_j}) \in \{1, 2, \dots, n\}^{r_j}$. In this way we will be able to specify a bijection f between the set of p-tuples of predicates $S_1, S_2, \dots, S_p$ of arities $r_1, r_2, \dots, r_p$, respectively, on $\{v_1, v_2, \dots, v_n\}$ and the set of the truth assignments for φ. If $\mathcal{S} = (S_1, S_2, \dots, S_p)$

is such a p-tuple of predicates, we define $f(\mathcal{S})$ as the following truth-value: *variable* $y^j_{i_1,i_2,\ldots,i_{r_j}}$ ***is true** iff* $(v_{i_1}, v_{i_2}, \ldots, v_{i_{r_j}}) \in S_j$. Once this bijection f defined, we will inductively construct φ so that the following property is preserved:

$$\mathcal{S} \models Q_1\left(x_1\right) Q_2\left(x_2\right) \ldots Q_r\left(x_r\right) \Phi\left(x_1, x_2, \ldots, x_r, \mathcal{P}, \mathcal{S}\right) \Longleftrightarrow f\left(\mathcal{S}\right) \models \varphi \qquad (1)$$

We start by eliminating quantifiers. For this, remark that, for any formula φ :

- $(\forall x \varphi(x, \mathcal{P}, \mathcal{S})) \Longleftrightarrow \varphi(x = v_1, \mathcal{P}, \mathcal{S}) \wedge \varphi(x = v_2, \mathcal{P}, \mathcal{S}) \wedge \ldots \wedge \varphi(x = v_n, \mathcal{P}, \mathcal{S})$;
- $(\exists x \varphi(x, \mathcal{P}, \mathcal{S})) \Longleftrightarrow \varphi(x = v_1, \mathcal{P}, \mathcal{S}) \vee \varphi(x = v_2, \mathcal{P}, \mathcal{S}) \vee \ldots \vee \varphi(x = v_n, \mathcal{P}, \mathcal{S})$.

In this way, we can, in r steps, transform formula $Q_1(x_1) \ldots Q_r(x_r) \Phi(x_1, x_2, \ldots, x_r, \mathcal{P}, \mathcal{S})$ into one consisting of n^r conjunctions or disjunctions of formulæ $\Phi(x_1 = v_{i_1}, x_2 = v_{i_2}, \ldots, x_r = v_{i_r}, \mathcal{P}, \mathcal{S})$. Formally, this new formula $\Psi'(\mathcal{P}, \mathcal{S})$ can be written as follows:

$$\bigodot_{i_1=1}^{n} \bigodot_{i_2=1}^{n} \ldots \bigodot_{i_r=1}^{n} \Phi\left(x_1 = v_{i_1}, x_2 = v_{i_2}, \ldots, x_r = v_{i_r}, \mathcal{P}, \mathcal{S}\right)$$

where the ith $\bigodot$ stands for $\vee$ if $Q_i = \exists$ and for $\wedge$ if $Q_i = \forall$.

Now, $\varphi = t(\Psi')$ is built by induction. If Ψ' is an elementary formula, then:

1. if $\Psi' = S_j(v_{i_1}, v_{i_2}, \ldots, v_{i_{r_j}})$, $\varphi = y^j_{i_1,i_2,\ldots,i_{r_j}}$;
2. if $\Psi' = P_j(v_{i_1}, v_{i_2}, \ldots, v_{i_{k_j}})$, $\varphi = \mathbf{true}$ if the instance is such that $(v_{i_1}, v_{i_2}, \ldots, v_{i_{k_j}}) \in P_j$ and **false** otherwise;
3. if Ψ' is formula $v_i = v_j$, then $\varphi = \mathbf{true}$ if $i = j$ and **false** otherwise.

Construction just described guarantees (1): in case 1, $\mathcal{S}$ verifies $\Psi'(\mathcal{P}, \mathcal{S})$ iff $(v_{i_1}, v_{i_2}, \ldots, v_{i_{r_j}}) \in S_j$, i.e., iff $y^j_{i_1,i_2,\ldots,i_{r_j}}\mathbf{true}$, therefore, iff $f(S)$ satisfies φ; in cases 2 and 3, either any $\mathcal{S}$ verifies Ψ', i.e., φ is a tautology, or no $\mathcal{S}$ verifies Ψ', i.e., φ is not satisfiable.

Assume now that Ψ' is non-elementary (i.e., composed by elementary formulæ); then,

- if $\Psi' = \neg\Psi''$, then $\varphi = t(\Psi') = \neg t(\Psi'')$;
- if $\Psi' = \Psi_1 \wedge \Psi_2$, then $\varphi = t(\Psi') = t(\Psi_1) \wedge t(\Psi_2)$;
- if $\Psi' = \Psi_1 \vee \Psi_2$, then $\varphi = t(\Psi') = t(\Psi_1) \vee t(\Psi_2)$.

Dealing with the first of items above:

$$\mathcal{S} \models \Psi' \iff \mathcal{S} \not\models \Psi'' \iff f(\mathcal{S}) \not\models t(\Psi'') \iff f(\mathcal{S}) \models \neg t(\Psi'')$$

For the second one (the third item is similar to the second one up to the replacement of "$\wedge$" by "$\vee$") we have:

$$\begin{aligned} \mathcal{S} \models \Psi' \iff \mathcal{S} \models \Psi_1 \wedge \mathcal{S} \models \Psi_2 &\iff f(\mathcal{S}) \models t(\Psi_1) \wedge f(\mathcal{S}) \models t(\Psi_2) \\ &\iff f(\mathcal{S}) \models t(\Psi_1) \wedge t(\Psi_2) \end{aligned}$$

We finally obtain a formula φ on $\sum_j n^{r_j}$ variables of size $n^r|\Phi|$. Furthermore, given (1), φ is obviously satisfiable iff $\exists \mathcal{S}\Psi(\mathcal{P}, \mathcal{S})$.

In general, φ is not CNF. We will build in polynomial time a CNF φ_S satisfiable iff φ does so. From so on, we assume that, when we define Π by $\exists \mathcal{S} Q_1(x_1) \dots Q_r(x_r)\Phi(x_1, \dots, x_r, \mathcal{P}, \mathcal{S})$, Φ is CNF.

Denote by $\varphi_b(i_1, i_2, \dots, i_r)$ the image with respect to t of $\Phi(x_1 = v_{i_1}, x_2 = v_{i_2}, \dots, x_r = v_{i_r}, \mathcal{P}, \mathcal{S})$. All these formulæ φ_b are, by construction, CNF and

$$\varphi = \bigodot_{i_1=1}^{n} \bigodot_{i_2=1}^{n} \dots \bigodot_{i_r=1}^{n} \varphi_b(i_1, i_2, \dots, i_r)$$

where the $\bigodot$ are as previously. Starting from φ we will construct, in a *bottom-up* way, formula φ_S in r steps (removing one quantifier per step). Note that if no quantifier does exist, then φ is CNF.

Suppose that q quantifiers remain to be removed. In other words, φ is satisfiable iff the following formula is satisfiable:

$$\bigodot_{i_1=1}^{n} \bigodot_{i_2=1}^{n} \dots \bigodot_{i_q=1}^{n} \left(C_1^{i_q} \wedge C_2^{i_q} \wedge \dots \wedge C_m^{i_q} \right)$$

where $C_i^{i_q}$ are disjunctions of literals.

If qth $\bigodot$ is $\wedge$, i.e., if qth quantifier is $\forall$, then$\wedge_{i_q=1}^{n}(C_1^{i_q} \wedge C_2^{i_q} \wedge \dots \wedge C_m^{i_q})$ is a conjunction of nm clauses, and consequently, we pass to $(q-1)$th quantifier.

If qth $\bigodot$ is $\vee$, things are somewhat more complicated. In this case, we define n new variables z^{i_q}, $i_q = 1, \dots, n$, and consider the following formula:

$$\varphi_q = \left(\bigvee_{i_q=1}^{n} z^{i_q} \right) \bigwedge \left(\bigwedge_{i_q=1}^{n} \left(z^{i_q} \Rightarrow \left(\bigwedge_{j=1}^{m} C_j^{i_q} \right) \right) \right)$$

Here, formula $\vee_{i_q=1}^{n} (C_1^{i_q} \wedge C_2^{i_q} \wedge \dots \wedge C_m^{i_q})$ is satisfiable iff formula φ_q does so. In fact,

- if a truth assignment satisfies the former, then for at least one q_0 conjunction of $C_j^{i_{q_0}}$ is true; then, we can extend this assignment by $z_{i_{q_0}} = \mathbf{true}$ and $z_{i_q} = \mathbf{false}$ if $q \neq q_0$;
- if a truth assignment satisfies φ_q, clause $(\vee_{i_q=1}^{n} z^{i_q})$ indicates that at least one $z_{i_{q_0}}$ is true; implication corresponding to this fact shows that conjunction of $C_j^{i_{q_0}}$ is true, and it suffices to restrict this truth assignment in order to satisfy formula $\vee_{i_q=1}^{n}$ $(C_1^{i_q} \wedge C_2^{i_q} \wedge \ldots \wedge C_m^{i_q})$.

Let us finally write φ_q in CNF. Note that:

$$z^{i_q} \Rightarrow \left(\bigwedge_{j=1}^{m} C_j^{i_q}\right) \equiv \left(\neg z^{i_q}\right) \vee \left(\bigwedge_{j=1}^{m} C_j^{i_q}\right) \equiv \bigwedge_{j=1}^{m} \left(\neg z_{i_q} \vee C_j^{i_q}\right)$$

In other words, $\neg z_{i_q} \vee C_j^{i_q}$ is a disjunction of literals. So, $\vee_{i_q=1}^{n} (C_1^{i_q} \wedge C_2^{i_q} \wedge \ldots \wedge C_m^{i_q})$ is satisfiable iff the following CNF formula is satisfiable:

$$\left(\bigvee_{i_q=1}^{n} z^{i_q}\right) \bigwedge \left(\bigwedge_{i_q=1}^{n} \bigwedge_{j=1}^{m} \left(\neg z^{i_q} \vee C_j^{i_q})\right)\right)$$

In all, we have added n new variables and constructed $1 + nm$ clauses. Obviously, construction described is polynomial. After r steps, we get a CNF φ_S satisfiable iff φ is satisfiable and overall construction is polynomial since each of its steps is polynomial (r does not depend on instance parameters). The proof of Cook's theorem is now complete.

Let us note that an analogous proof has pointed out to us after having accomplished what it has just presented. It is given by Immerman in [4]. Immerman's proof is quite condensed, and based upon another version of Fagin's theorem. Furthermore, the type of reduction used, called *first-order reduction*, is, following the author, weaker than classical Karp's reduction. This is not the case of our proof which, to our opinion, is a Karp's reduction.

2 Constructing CNFs for MAX INDEPENDENT SET and 3-COLORING

2.1 MAX INDEPENDENT SET

An instance of MAX INDEPENDENT SET consists of a graph $G(V, E)$, with $|V| = n$ and $|E| = m$, and an integer K. The question is if there exists a set $V' \subseteq V$, with

$|V'| \geqslant K$ such that no two vertices in V' are linked by an edge. The most natural way of writing this problem as a logical formula is the following:

$$\begin{aligned} \exists S \qquad & \forall x \forall y\, (S(x) \wedge S(y)) \Rightarrow \neg G(x,y) \\ & \wedge \exists y_1 \exists y_2 \neq y_1 \ldots \exists y_K \neq y_1 \ldots y_{K-1} S\,(y_1) \wedge S\,(y_2) \ldots \wedge S\,(y_K) \end{aligned}$$

However, in this form the number of quantifiers depends on K, therefore on problem's instance and transformation of Section 1 is no more polynomial. In order to preserve polynomiality of transformation, we are going to express MAX INDEPENDENT SET a problem of determining a permutation P on the vertices of G such that the the K first vertices of P form an independent set. Consider a predicate S of arity 2 such that $S(v_i, v_j)$ iff $v_j = P[v_i]$. MAX INDEPENDENT SET can be formulated as follows (in this formulation, $v_i \leqslant v_j$ means $i \leqslant j$) :

$$\begin{aligned} \exists S \qquad & \Big(\forall x \exists y S(x,y)\Big) \wedge \Big(\forall x \forall y \forall z \big(S(x,y) \wedge S(x,z)\big) \Rightarrow y = z\Big) \\ & \wedge \Big(\forall x \forall y \forall z \big(S(x,z) \wedge S(y,z)\big) \Rightarrow x = y\Big) \\ & \wedge \Big(\forall x \forall y \forall z \forall t \big(x \neq y \wedge S(x,z) \wedge S(y,t) \wedge z \leqslant v_K \wedge t \leqslant v_K\big) \Rightarrow \neg G(x,y)\Big) \end{aligned}$$

Here, first line expresses the fact that predicate S represents a function of the vertex-set in itself, the second one that this function is injective (consequently, bijective also) ; finally, third line indicates that the K first vertices $(v_1, \ldots, v_K)$ of $P[V]$ form an independent set. Formula above is rewritten in prenex form as follows:

$$\begin{aligned} \exists S\ \forall x \forall y \forall z \forall t \exists u \qquad & S(x,u) \wedge \Big(\neg S(x,y) \vee \neg S(x,z) \vee y = z\Big) \\ & \wedge \Big(\neg S(x,z) \vee \neg S(y,z) \vee x = y\Big) \\ & \wedge \Big(x = y \vee \neg S(x,z) \vee \neg S(y,t) \vee z > v_K \vee t > v_K \vee \neg G(x,y)\Big) \end{aligned}$$

We construct a CNF on $n^2 + n$ variables: n^2 variables $y_{i,j}$ representing the fact that $(v_i, v_j) \in S$, and n variables z^i because the last quantifier is existential. We so get the following clauses:

- clause $z^1 \vee z^2 \ldots \vee z^n$ coming from removal of the existential quantifier;
- n^2 clauses: $\bar{z}^j \vee y_{i,j}$ ($i = 1, \ldots, n$, $j = 1, \ldots, n$);
- $\forall (i,j,k,l) \in \{1, \ldots, n\}^4$ where $k \neq l$, clause $\bar{z}^j \vee \bar{y}_{i,k} \vee \bar{y}_{i,l}$;
- $\forall (i,j,k,l) \in \{1, \ldots, n\}^4$ where $i \neq k$, clause $\bar{z}^j \vee \bar{y}_{i,l} \vee \bar{y}_{k,l}$;
- $\forall (i,j,k,l,m) \in \{1, \ldots, n\}^5$ where $i \neq k$, $l \leqslant K$ and $m \leqslant K$ is such that edge $(v_i, v_k) \in E$, clause $\bar{z}^j \vee \bar{y}_{i,l} \vee \bar{y}_{k,m}$.

We so obtain a formula on n^2+n variables with at most $mnK^2+2(n^4-n^3)+n^2+1 \leqslant O(n^5)$ clauses.

2.2 3-COLORING

A graph G of order n is 3-colorable if there exists S_1, S_2, et S_3 such that:

$$\begin{aligned}\forall x \forall y \quad & \Big(S_1(x) \vee S_2(x) \vee S_3(x)\Big) \wedge \Big(\neg S_1(x) \vee \neg S_2(x)\Big) \wedge \Big(\neg S_1(x) \vee \neg S_3(x)\Big) \\ & \wedge \Big(\neg S_2(x) \vee \neg S_3(x)\Big) \wedge \Big(\neg G(x,y) \vee \neg S_1(x) \vee \neg S_1(y)\Big) \\ & \wedge \Big(\neg G(x,y) \vee \neg S_2(x) \vee \neg S_2(y)\Big) \wedge \Big(\neg G(x,y) \vee \neg S_3(x) \vee \neg S_3(y)\Big)\end{aligned}$$

Remark that the formula above is the CNF equivalent of the 3-COLORING formula seen in Section 1. Formula φ_S is then defined on:

- $3n$ variables y_i^j, $j = 1, \dots, 3$ and $i = 1, \dots, n$; $y_i^j = \mathbf{true}$ if v_i receives color j;
- n series of clauses $(y_i^1 \vee y_i^2 \vee y_i^3) \wedge (\bar{y}_i^1 \vee \bar{y}_i^2) \wedge (\bar{y}_i^1 \vee \bar{y}_i^3) \wedge (\bar{y}_i^2 \vee \bar{y}_i^3)$ (where i goes from 1 to n); series corresponding to index i represents the fact that vertex v_i receives one and only one color;
- clauses representing constraints on adjacent vertices, i.e., for any edge (v_i, v_j) of G, v_i and v_j are colored with different colors: $(\bar{y}_i^1 \vee \bar{y}_j^1) \wedge (\bar{y}_i^2 \vee \bar{y}_j^2) \wedge (\bar{y}_i^3 \vee \bar{y}_j^3)$;

We so get a CNF on $3n$ variables with $4n + 3m$ clauses, any clause containing at most 3 literals.

3 The Min NPO-completeness of MIN WSAT

In MIN WSAT, we are given a CNF φ on n variables $x_1, \dots, x_n$ and m clauses $C_1, \dots, C_m$. Any variable x_i has a non-negative weight w_i, $i = 1, \dots, n$. We assume that the assignment $x_i = 1$, $i = 1, \dots, n$ is a feasible solution, and we denote it by $\mathrm{triv}(\varphi)$. The objective of MIN WSAT is to determine an assignment $T = (t_1, \dots, t_n)$, $t_i \in \{0, 1\}$, on the variables of φ in such a way that (i) T is a model for φ and (ii) quantity $\sum_{i=1}^n t_i w_i$ is minimized.

Always based upon Fagin's characterization of **NP**, we show in this section the **Min NPO**-completeness of MIN WSAT under a kind of approximation preserving reduction, originally defined in [5], called *strict reduction*. The class **Min NPO** is the class of minimization **NPO** problems. An optimization problem is in **NPO** if its decision version is in **NP** (see [1] for more details about definition of **NPO**). More formally, an **NPO** problem Π is defined as a four-tuple $(\mathcal{I}, \mathrm{sol}, m, \mathrm{opt})$ such that: $\mathcal{I}$ is the set of instances of Π and it can be recognized in polynomial time; given $x \in \mathcal{I}$, $\mathrm{sol}(x)$ denotes the set of feasible solutions of x; for every $y \in \mathrm{sol}(x)$, $|y|$ is polynomial in $|x|$; given any x and any y polynomial in $|x|$, one can decide in polynomial time if $y \in \mathrm{sol}(x)$; given $x \in \mathcal{I}$ and $y \in \mathrm{sol}(x)$, $m(x, y)$ denotes the value of y for x; m is polynomially computable and is commonly called feasible value; finally, $\mathrm{opt} \in \{\max, \min\}$. We assume that any instance x of any **NPO** problem admits at least one feasible solution, denoted by $\mathrm{triv}(x)$, computable in polynomial time.

Given an instance x of Π, we denote by $\mathrm{opt}(x)$ the value of an optimal solution of x. For an approximation algorithm A computing a feasible solution y for x with value $m_{\mathtt{A}}(x, y)$, its approximation ratio is defined as $r_{\Pi}^{\mathtt{A}}(x, y) = m_{\mathtt{A}}(x, y)/\operatorname{opt}(x)$.

Consider two **NPO** problems $\Pi = (\mathcal{I}, \mathrm{sol}, m, \mathrm{opt})$ and $\Pi' = (\mathcal{I}', \mathrm{sol}', m', \mathrm{opt})$. A *strict reduction* is a pair (f, g) of polynomially computable functions, $f : \mathcal{I} \to \mathcal{I}'$ and $g : \mathcal{I} \times \mathrm{sol}' \to \mathrm{sol}$ such that:

- $\forall x \in \mathcal{I}, x \mapsto f(x) \in \mathcal{I}'$;
- $\forall y \in \mathrm{sol}'(f(x)), y \mapsto g(x, y) \in \mathrm{sol}(x)$;
- if r is an approximation measure, then $r_{\Pi}(x, g(x, y))$ is as good as $r_{\Pi}(f(x), y)$.

Completeness of MIN WSAT has been originally proved in [5], based upon an extension of Cook's proof ([2]) of SAT **NP**-completeness to optimization problems. As we have already mentioned just above, we give an alternative proof of this result, based upon Fagin's characterization of **NP**.

3.1 Construction of f

Consider a problem $\Pi = (\mathcal{I}, \mathrm{sol}, m, \min)$ and denote by $m(x, y)$ the value of solution y for instance $x \in \mathcal{I}$, set $n = |x|$ and assume two polynomials p and q such that, $\forall x \in \mathcal{I}, \forall y \in \mathrm{sol}(x), 0 \leqslant |y| \leqslant q(n)$ and $0 \leqslant m(x, y) \leqslant 2^{p(n)}$. As in the proof of [5], we define the following Turing-machine M:

```
Turing machine $M$
on input $x$ :
if $x \notin \mathcal{I}$, then reject;
generate a string $y$ such that $|y| \leqslant q(n)$;
if $y \notin \mathrm{sol}(x)$, then reject;
write $y$;
write $m(x,y)$ ;
accept.
```

By the proof of Fagin's theorem ([3]), one can construct a second-order formula $\exists S\Phi(S)$ satisfiable iff M accepts x. Revisit this proof for a while; it consists of writing, for an instance x, table $\mathcal{M}$ of $M_x(i,j)$, where $M_x(i,j)$ represents the symbol written at instant i in th jth entry of M (when running on x). If M runs in time n^k, then i and j range from 0 to $n^k - 1$. Second-order formula is then built in such a way that it describes the fact that, for an instance x, there exists such a table $\mathcal{M}$ corresponding to both the way M functions and to the fact that M arrives to acceptance in time $n^k - 1$. Consider that machine's alphabet is $\{0, 1, b\}$, where b is the blank symbol and suppose that when M arrives in acceptance state there is no further changes; this implies that when M attains acceptance state, one can read results of computation on line of $\mathcal{M}$ corresponding to instant $n^k - 1$.

What is of interest for us in Fagin's proof is predicates $S_0(t,s)$ and $S_1(t,s)$ representing the fact that 0, or 1, are written at instant i (encoded by t) on tape-entry j (encoded by s); t and s are two k-tuples $t_1, t_2, \ldots, t_k$ and $s_1, s_2, \ldots, s_k$ of values in $\{0, n-1\}$. An integer $i \in \{0, n^k - 1\}$ written to the base n can be represented by a k-tuple $t_1, t_2, \ldots, t_k$ in such a way that $i = \sum_{l=1}^{k} t_i n^{i-1}$. In what follows $b(t)$ will denote the value whose $t = (t_1, \ldots, t_k)$ is the representation to the base n ($b(t) = \sum_{l=1}^{k} t_i n^{i-1}$). Predicates S_0 and S_1 allow recovering of value computed by M since this value is written on line corresponding to instant $n^k - 1 = b(t_{\max})$, with $t_{\max} = (n-1, \ldots, n-1)$.

By the way M is defined, in case of accepting computation, on the last line of the corresponding table $\mathcal{M}$, solution y and its value $m(x,y)$ are written. Denote by $c_0, c_1, \ldots, c_{p(n)}$ the entries of M where $m(x,y)$ is written (in binary). This value is:

$$m(x,y) = \sum_{j:c_j=1} 2^j = \sum_{j:\left\{\substack{c_j=b(s) \\ S_1(t_{\max},s)}\right.} 2^j$$

We now transform second-order formula in Fagin's theorem into an instance of SAT as described in Section 1. Among other ones, this formula contains variables $y^1_{t,s}$ "representing" predicate S_1 of arity $2k$ (with $t = (t_1, \ldots, t_k)$, $s = (s_1, \ldots, s_k)$, where t_i and s_i, $i = 1, \ldots, k$, range from 0 to $n-1$). Denote by φ the instance of SAT so-obtained and assume the following weights on variables of φ:

$$\begin{cases} w\left(y^1_{t,s}\right) = 2^j & \text{if } t = t_{\max} \text{ and } c_j = b(s) \\ w(y) = 0 & \text{otherwise} \end{cases}$$

In other words, we consider weight 2^j for variable representing the fact that entry c_j contains an 1.

We so obtain an instance of MIN WSAT and the specification of component f of strict reduction (f, g) transforming an instance of any **NPO** problem Π into an instance φ of MIN WSAT is complete.

3.2 Construction of g

Consider now an instance x de Π and a feasible solution z of $\varphi = f(x)$. Define component g of the reduction as:

$$g(x,z) = \begin{cases} \operatorname{triv}(x) & \text{if } z = \operatorname{triv}(f(x)) \\ \text{the solution accepted by } M & \text{otherwise} \end{cases}$$

Solution accepted by M and its value can both be recovered, as we have discussed, using predicates S_0 and S_1 (recall that truth values of these predicates are immediately deduced from z by the relation "$S_i(t, s)$ iff $y^i_{t,s} = \mathbf{true}$", $i \in \{0, 1\}$). Specification of g is now complete.

3.3 Reduction (f, g) is strict

The pair (f, g) specified above constitute a reduction of Π to MAX SAT. It remains to show that this reduction is strict. We distinguish the following two cases:

- if $z = \operatorname{triv}(f(x))$, then $y = g(x, z) = \operatorname{triv}(x)$; in this case:

  $$m(x,z) \leqslant \sum_{j=0}^{p(n)} 2^j = w(z)$$

  where by $w(z)$ we denote the total weight of solution z;

- otherwise, $y = g(x, z)$ and, by construction:

  $$m(x,y) = \sum_{j:c(j)=1} 2^j = \sum_{j:\begin{cases} c_j=b(s) \\ S_1(t_{\max},s) \end{cases}} 2^j = \sum_{j:\begin{cases} c_j=b(s) \\ y^1_{t_{\max},s}=\mathbf{true} \end{cases}} 2^j = w(z)$$

Since optimal solution-values of instances x and $f(x)$ are also equal, so do approximation ratios. Therefore reduction specified above is strict.

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and approximation. Combinatorial optimization problems and their approximability properties*. Springer, Berlin, 1999.

[2] S. A. Cook. The complexity of theorem-proving procedures. In *Proc. STOC'71*, pages 151–158, 1971.

[3] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In R. M. Karp, editor, *Complexity of computations*, pages 43–73. American Mathematics Society, 1974.

[4] N. Immerman. *Descriptive complexity*. Springer-Verlag, 1998.

[5] P. Orponen and H. Mannila. On approximation preserving reductions: complete problems and robust measures. Technical Report C-1987-28, Dept. of Computer Science, University of Helsinki, Finland, 1987.

On-line models and algorithms for MAX INDEPENDENT SET

Bruno Escoffier[*], Vangelis Th. Paschos[*]

Résumé

Dans un problème on-line, l'instance du problème n'est pas entièrement connue au départ mais est révélée étape par étape, le but étant de construire, au cours de ce processus où l'on découvre l'instance, une solution réalisable la meilleure possible. Nous nous intéressons dans cet article au problème du stable on-line. Les modèles étudiés jusqu'à présent pour le problème du stable on-line consistent, partant d'un graphe vide, à révéler le graphe sommet par sommet, ou par sous ensembles de sommets. Nous allons ici nous intéresser à deux nouveaux modèles du stable on-line. Premièrement, nous étudierons l'approximabilité du problème lorsque le graphe est initialement complet et des arêtes disparaissent à chaque étape, jusqu'à l'obtention du graphe final. Ensuite, nous reprenons les modèles étudiés dans [M. Demange, X. Paradon and V. Th. Paschos, *On-line maximum-order induced hereditary subgraph problems*, Proc. SOFSEM 2000—Theory and Practice of Informatics, LNCS 1963, pp. 326–334, 2000] et proposons une relaxation basée sur l'introduction d'une possibilité de *backtracking* payant.

Mots-clefs : Algorithmique on-line; Ensemble stable, Rapport de compétitivité

Abstract

In on-line computation, instance of the problem dealt is not entirely known from the beginning of the solution process, but it is revealed step-by-step. In this paper we deal with on-line independent set. On-line models studied until now for this problem suppose that the input graph is initially empty and revealed either vertex-by-vertex, or cluster-by-cluster. Here we present a new on-line model quite different to the ones already studied. It assumes that a superset of the final graph is initially present (in our case the complete graph on the order n of the final graph) and edges are progressively

[*] LAMSADE, Université Paris-Dauphine, 75775 Paris cedex 16, France. {escoffier,paschos}@lamsade.dauphine.fr

removed until the achievement of the final graph. Next, we revisit model introduced in [M. Demange, X. Paradon and V. Th. Paschos, *On-line maximum-order induced hereditary subgraph problems*, Proc. SOFSEM 2000—Theory and Practice of Informatics, LNCS 1963, pp. 326–334, 2000] and study relaxations assuming that some paying backtracking is allowed.

Key words : Approximation algorithms; Competitive ratio; Maximum independent set; On-line algorithms

1 Introduction

1.1 On-line computation

On-line algorithms have been introduced to tackle situations where problem's solution is planned under uncertainty concerning the final instance of the problem dealt. This kind of situations appears frequently when we have to efficiently solve a problem in real time. In such situations we need to start problem's resolution before the whole instance is completely known. They lead to what is called on-line combinatorial optimization problems. Models for such problems are usually based upon the following two constitutive hypotheses:

1. instance of the problem is revealed step-by-step;
2. decision makers make choices once a part of the instance is revealed, these choices being definite and irrevocable.

Starting from these hypotheses, one can built different models depending on how instance is precisely revealed and what are the rights of the decision maker (on-line algorithm) for constructing the final solution.

Since about twenty years, many combinatorial optimization problems have been studied in on-line versions. For example, [1] studies on-line models for TRAVELLING SALESMAN, [3, 9] study models for on-line MAX INDEPENDENT SET, etc. Also, an interesting survey about on-line combinatorial optimization problems can be found in [10]. In fact, what is easily understood from all these papers, is that on-line-computation is a natural extension of approximation theory.

Given an on-line problem Π, an on-line algorithm for Π provides, for any instance x (and following to the rules of the on-line model describing Π) a feasible solution y for x. The quality of y is measured by the so-called competitive ratio $m(x,y)/\mathrm{opt}(x)$, where $m(x,y)$ is the value of y and $\mathrm{opt}(x)$ the value of the optimal off-line solution for x. We will say that an on-line algorithm A guarantees competitive ratio $f(x)$, if f is a function such that, for any instance x of Π the competitive ratio of solution y computed by A

is better (greater than, or equal to, if we deal with a maximization problem, less than, or equal to, if the problem dealt is a minimization one) than $f(x)$.

1.2 On-line models for MAX INDEPENDENT SET

Given a graph $G(V,E)$, an *independent set* is a subset $V' \subseteq V$ such that whenever $\{v_i,v_j\} \subseteq V'$, $v_iv_j \notin E$, and MAX INDEPENDENT SET consists in finding an independent set of maximum size. In weighted MAX INDEPENDENT SET we consider that vertices are provided with positive weights and the objective becomes to determine an independent set of maximum total weight.

For MAX INDEPENDENT SET, the most natural on-line model seems to be the following one: the initial graph is empty and vertices are revealed one-by-one; together with a "new" vertex all edges linking it with "old" ones are simultaneously revealed. Once a new vertex arrives an algorithm for this model has to decide if this vertex will be included in the solution under construction or not. Such a model is however quite restrictive since no on-line algorithm can guarantee competitive ratio better than $1/(n-1)$, where n is the order of the final graph (while any on-line algorithm trivially achieves this ratio). In [3] a relaxation of this model is proposed. There, instead vertex-by-vertex, G is revealed within $t < n$ clusters. Any time a new cluster arrives all edges linking its vertices with the ones of the older clusters are also revealed. Any on-line algorithm has then to decide which among the vertices of this new cluster have to be integrated in the independent set under construction. It is proved there that, if t clusters are needed that the whole graph is revealed, there exists a polynomial time on-line algorithm achieving competitive ratio $\Omega(\log n/(n\sqrt{t}))$. Two other kinds of relaxations of the basic model are studied in [9]. There, it is assumed that the algorithm can maintain a collection of n^k independent sets (for some constant k) and at the end of the game it can choose the best of the solutions maintained. Under this assumption a competitive ratio $\Omega(\log n/n)$ is achieved. In the second model of [9], the algorithm is allowed to copy intermediate solutions and to extend the copied solutions in different ways. The so-obtained competitive ratio is, once again, $\Omega(\log n/n)$.

Recall that the best known approximation ratios for MAX INDEPENDENT SET and WEIGHTED MAX INDEPENDENT SET are:

- (asymptotical) k/Δ [4] and $3/(\Delta+2)$ [7], respectively,
- $\Omega(\log^2 n/n)$ [8] (for both versions),
- $\min\{\Omega(\log n/(\Delta \log\log n)), n^{-4/5}\}$ [5] (for both versions also).

In this paper we first study (Section 2) the following on-line MAX INDEPENDENT SET-model: the initial graph is a clique on n vertices and in each step some (one or more) of its edges are removed; any time edges are removed, the on-line algorithm is allowed to add to the independent set under construction vertices adjacent to some of these edges.

Next, in Section 3, we revisit the on-line model already studied in [3] and we relax it by ignoring irrevocability requirement, assuming instead that any time a decision is changed, this change is charged by some non-negative cost. Note that, relaxations dealing with decisions irrevocability also appears in [9], where algorithm is allowed to maintain at each step several solutions in order to finally return the best among them.

In what follows, we denote by n the order of the input-graph G, by Δ its maximum degree and by $\alpha(G)$ the cardinality of a maximum independent set for G (commonly called stability or independence number [2]).

2 On-line edge removal

As we have already mentioned, the model studied in this section consists of starting from a complete graph on n vertices (this is also the order of the final graph) and of supposing that edges disappear step-by-step; at each step, one or more edges are removed. Upon the removal of a set of edges, algorithm has to irrevocably decide which of the vertices adjacent to them are included to the independent set under construction.

We first suppose that the number of steps (iterations) needed that the final graph is fixed is not known in advance to the algorithm (section 2.1). In this case we propose a natural greedy on-line algorithm and show that it is strongly competitive. Next, we suppose that number of iterations needed for fixing the final graph is known in advance. For this case, we devise an on-line algorithm achieving non-trivial competitive ratio in particular when the final graph is fixed within a small number of iterations. For any of the cases dealt we also prove upper bounds to the corresponding competitive ratios.

2.1 The number of iterations is not known in advance

Denote by GA, the natural greedy MAX INDEPENDENT SET algorithm (see, for example, [11] for more details about its approximability). The on-line algorithm considered here is the following, denoted by OLGA:

- at step i, determine the subgraph H_i, induced by the vertices that are adjacent to the edges just removed but non-adjacent to the vertices already included in the independent set S under construction;
- compute $\mathtt{GA}(H_i)$;
- solution at step i becomes $S = S \cup \mathtt{GA}(H_i)$.

Proposition 1 *Competitive ratio achieved by OLGA is bounded below by $2/(n-1)$, if the parameter dealt for the analysis is the order n of the input graph, or $1/\Delta$, if the parameter dealt is the maximum graph-degree Δ.*

Proof. Assume first that the parameter for the analysis of competitiveness is n. We distinguish two cases, namely, $\alpha(G) = n$ and $\alpha(G) < n$. For the former one, the final graph is simply a set of isolated vertices and there OLGA trivially determines an independent set of cardinality n, achieving so a competitive ratio $1 > 2/(n-1)$. For the latter case, obviously the final graph contains at least one edge; so, $\alpha(G) \leqslant n-1$. Here, OLGA will determine an independent set containing at least the endpoints of a removed edge, i.e., an independent set of cardinality at least 2. The claimed ratio is so proved.

Assume now that the parameter for the analysis of competitiveness is Δ and note that OLGA always computes an independent set maximal for the inclusion. Such an independent set S always garanties $|S|/\alpha(G) \geqslant 1/\Delta$ [11]. ∎

Theorem 1 *No on-line algorithm can achieve competitive ratio strictly greater than* $2/(n-1)$ *or* $1/\Delta$ *(under the on-line model assumed) for* MAX INDEPENDENT SET.

Proof. Denote by A an on-line algorithm for MAX INDEPENDENT SET constructing an independent set S and run it in the following instance:

- during the first iteration, edge (v_1,v_2) is removed;
- if A does not choose any of v_1 or v_2, game is over;
- otherwise, A has chosen at least one among v_1 and v_2, say v_1;
- then, the second iteration consists of removing all edges in the subgraph of G induced by vertex-set $\{v_2,\ldots,v_n\}$.

In the case where A has not made any choice among v_1 and v_2, we have $|S| = 0$ and $\alpha(G) = 2$. In the case where at least v_1 has been introduced in S, no vertex in $\{v_3,\ldots,v_n\}$ can complete it (since all these vertices are linked to v_1). Hence, $|S| \leqslant 2$, while maximum independent set is $\{v_2,\ldots,v_n\}$ of cardinality $n-1$. In both cases, $|S|/\alpha(G) \leqslant 2/(n-1)$, and the proof of the first statement of the theorem.

Fix now a $\Delta > 2$. We will build a graph G of maximum degree Δ and of order $n = k(\Delta+1)$ (for any $k \geqslant 2$). Group the n vertices $v_1,v_2,\ldots,v_n$ in k groups of $\Delta+1$ vertices per group (assuming that $v_1,v_2,\ldots,v_{\Delta+1}$ are put in the first group, …, $v_{(k-1)(\Delta+1)+1},\ldots,,v_{k(\Delta+1)}$ are in the kth (last) group).

First iteration consists of removing all edges linking two vertices lying to two distinct groups. We so obtain a non-connected graph on k connected components $G_1,G_2,\ldots,G_k$ each of these components been a complete subgraph on $\Delta+1$ vertices. Here A can choose at most one vertex per graph G_i, $i = 1,\ldots,k$ to add it in S. Suppose that it chooses a vertex in each of the graphs $G_1,G_2,\ldots,G_l$, $0 \leqslant l \leqslant k$, and no vertex in the rest of the components. In the second and last iteration we remove in $G_1,\ldots,G_l$ any edge non-incident to the vertex chosen by A. The form of the graph is as shown in figure 1, assuming $\Delta = 5$, $k = 5$, $l = 3$, as well as that white vertices have been added in S.

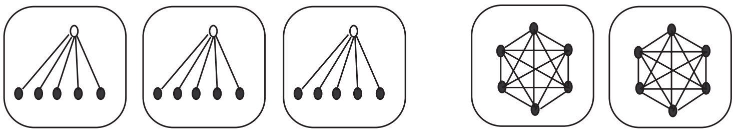


FIG. 1 – $\Delta = 5$, $k = 5$, $l = 3$

In this case, no vertex can be added in S, since all vertices incident to the edges just removed are linked to the vertex added previously; hence, $|S| = l$. On the other hand, there exists an independent set S^* with $|S^*| = l\Delta + (k - l)$ in G containing

- the Δ vertices linked to the vertex chosen by A in any of G_1,$G_2, \ldots$,G_l and
- one vertex per graph $G_{l+1}, \ldots$,G_k.

So, $|S|/\alpha(G) \leqslant l/(l\Delta + k - l) \leqslant 1/\Delta$, which completes the proof of the second statement and of the theorem. ▮

Proposition 1 and Theorem 1 immediately conclude the following corollary.

Corollary 1 *Algorithm OLGA is optimally competitive for* MAX INDEPENDENT SET *(under the model dealt).*

With very similar arguments the following upper bounds can be proved for the case where the final graph is connected.

Theorem 2 *Assuming that the final graph is connected,*

- *no on-line algorithm can achieve competitive ratio better than $2/(n - 2)$, for any graph of order $n \geqslant 4$;*
- *no on-line algorithm can achieve competitive ratio better than $1/(\Delta - 1)$.*

Furthermore, one can easily prove that OLGA achieves competitive ratios $2/(n - 2)$ and $1/\Delta$ on connected graphs, but not $1/(\Delta - 1)$.

2.2 The number of iterations is known in advance

We assume in this section that number of steps, denoted by t, needed for revealing the graph is known in advance, i.e., it is, in some sense, part of the instance of the on-line MAX INDEPENDENT SET. We also use the following notations: G_i denotes the graph at the end of iteration i; hence, $G_t = G$ and, since in any iteration we remove some edges, G is a partial subgraph of G_i for any $i \leqslant t$; I^i denotes the set of vertices adjacent to the set of edges removed during iteration i, i.e., the set of vertices that can be added to the solution under construction in iteration i. Moreover, if V' is a subset of the vertex-set V

of G, $G_i[V']$ denotes the subgraph of G_i induced by V'. For example, $G_t[I^1]$ corresponds to the subgraph of $G_t = G$ induced by the vertices one could add in the solution during the first iteration.

Let us consider, for example, figure 2. The final graph contains 5 vertices $\{1,2,\ldots,5\}$ and is revealed in three steps. During first iteration, edges $(1,4)$ and $(2,3)$ disappear; so, $I^1 = \{1,2,3,4\}$. During second iteration edges $(2,5)$ and $(3,4)$ are further removed, and $I^2 = \{2,3,4,5\}$. Finally, during third (last) iteration, edges $(1,3)$ is also removed; hence, $I^3 = \{1,3\}$. In figure 3, graphs $G_1[I^1]$, $G_2[I^1]$, and $G_3[I^2]$ are illustrated.

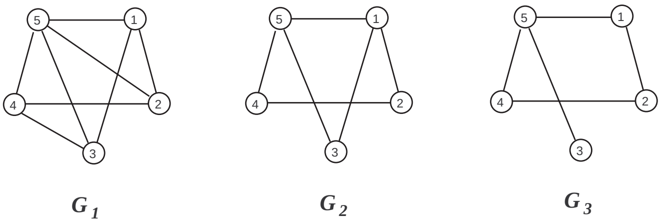


FIG. 2 – *The three steps for revealing G.*

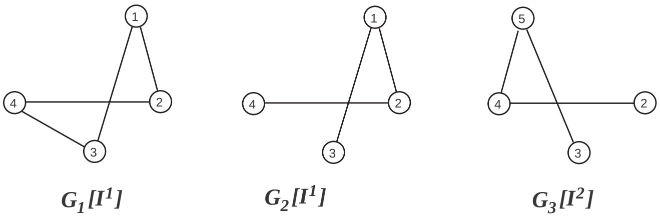


FIG. 3 – *Graphs $G_1[I^1]$, $G_2[I^1]$, and $G_3[I^2]$*

The following lemma gives an upper bound for $\alpha(G)$ linking it to quantities $\alpha(G_i[I^i])$.

Lemma 1 $\alpha(G) \leq \sum_{i=1}^{t} \alpha(G_i[I^i])$, *and this independently on the way G is revealed.*

Proof. Let J^i be the subset of I^i that does not belong to any I^k, $k > i$; J^i corresponds to the vertices that one can choose to put in the solution for last time during iteration i (obviously, J^i can be empty). For example, in figure 2, $J^1 = \emptyset$, $J^2 = \{2,4,5\}$ and $J^3 = \{1,3\}$.

Let S^* be a maximum independent set of G. Remark that vertices of G that do not belong to any J^i are exactly those that are not adjacent to any edge removed; hence they are linked to any other vertex of G and, consequently, they cannot belong to S^*. So, sets $S^* \cap J^i$, $1 \leqslant i \leqslant t$ form a partition on S^* (note that some of these sets may be empty).

On the other hand, set $S^* \cap J^i$ is an independent set of G and is included in I^i (because $J^i \subset I^i$). Note also that an independent set of G included in I^i is not mandatorily an independent set of $G_i[I^i]$. In fact, no reason forbids that edges linking two vertices of I^i are removed during an iteration subsequent to iteration i; for example, in figure 2, set $\{1,3\}$ is an independent set of G but not of $G_1[I^1]$. But under the definition of J^i just above, $S^* \cap J^i$ is indeed an independent set for $G_i[I^i]$. We so obtain: $\alpha(G) = |S^*| = \sum_{i=1}^{t} |S^* \cap J^i| \leqslant \sum_{i=1}^{t} \alpha(G_i[I^i])$, qed. ∎

Consider now the following algorithm for on-line MAX INDEPENDENT SET, denoted by OLTA calling as subroutine an approximation MAX INDEPENDENT SET-algorithm AA achieving approximation ratio $\rho(n)$ ($r(n,t)$ is a threshold to be precised later and $\rho(n)$ decreases with n):

- set $i = 1$;
- while $|\mathtt{AA}(G_i[I^i])| < r(n,t)$ and $i < t$, set $i = i + 1$;
- output $S = \mathtt{AA}(G_i[I^i])$.

Theorem 3 *Algorithm OLTA achieves competitive ratio $\sqrt{\rho(n)/(nt)}$ for on-line* MAX INDEPENDENT SET *(under the model considered). Moreover, it is polynomial if A is so.*

Proof. If OLTA outputs a solution before iteration t, then the size of this solution is $|\mathtt{AA}(G_i[I^i])| \geqslant r(n,t)$ and the competitive ratio thus achieved is at least

$$\frac{r(n,t)}{\alpha(G)} \geqslant \frac{r(n,t)}{n} \tag{1}$$

Otherwise, for all $i \in \{1,2,\cdots,t\}$,

$$\left|\mathtt{AA}\left(G_i\left[I^i\right]\right)\right| < r(n,t) \tag{2}$$

Since AA is assumed to guarantee approximation ratio ρ,

$$\frac{|\mathtt{AA}\left(G_i\left[I^i\right]\right)|}{\alpha\left(G_i\left[I^i\right]\right)} \geqslant \rho\left(\left|I^i\right|\right) \geqslant \rho(n) \tag{3}$$

Using (2), (3) and Lemma 1, we get:

$$\alpha(G) \leqslant \sum_{i=1}^{t} \alpha\left(G_i\left[I^i\right]\right) \leqslant \sum_{i=1}^{t} \frac{|\mathtt{AA}\left(G_i\left[I^i\right]\right)|}{\rho(n)} \leqslant \sum_{i=1}^{t} \frac{r(n,t)}{\rho(n)} = \frac{r(n,t)t}{\rho(n)}$$

and consequently,

$$\frac{|\mathtt{OLTA}(G)|}{\alpha(G)} \geqslant \frac{1}{\alpha(G)} \geqslant \frac{\rho(n)}{r(n,t)t} \tag{4}$$

Ratio in (1) is increasing with $r(n,t)$, while the one in (4) is decreasing with $r(n,t)$. Equality of them holds for $r(n,t) = \sqrt{\frac{n\rho(n)}{t}}$ and, in this case, ratio achieved is as claimed. ▮

Corollary 2 *Dealing with an optimal off-line algorithm* AA*, competitive ratio implied by Theorem 3 is $1/\sqrt{nt}$. If, on the other hand,* AA *is the polynomial time approximation algorithm of [8], then the competitive ratio achieved by* OLTA *is bounded below by $\Omega(\log n/(n\sqrt{t}))$. In particular, when t is fixed, then this ratio is $\Omega(\log n/n)$.*

Theorem 4 *No on-line algorithm can achieve, for the on-line* MAX INDEPENDENT SET*-model considered, competitive ratio strictly better than $1/(\sqrt{n/2} - 1)$ (for $n \geqslant 3$), even if $t = 2$.*

Proof. Consider an on-line algorithm A, an integer $n \geqslant 3$ and set $p = \lfloor\sqrt{2n}\rfloor$ ($p < n$). Assume that first step of graph revealing consists of removing edges in such a way that set $V' = \{v_1,v_2,\ldots,v_p\}$ becomes an independent set; assume also that only such edges are removed during this first step. For the second step we distinguish the two following cases:

1. if, during first iteration, A has chosen at least a vertex (vertices chosen belong to V'), then, in the second step, we remove all edges non-incident to any vertex in V'; in such a case A cannot extend independent set previously constructed while there exists in G an independent set of size $n - p$ composed by all vertices in $V \setminus V'$; so, competitive ratio of A is, in this case, at most $p/n - p$;
2. otherwise, in second step we remove only one edge, say (v_1,v), where v can be any vertex not in V'; in this case, the independent set built by A contains at most two vertices (v_1 and v), while set $V' \cup \{v\}$ is an independent set for G of size $p + 1$; consequently, the competitive ratio achieved by A is at most $2/(p + 1)$.

Combination of the ratios of Cases 1 and 2, together with the fact that $p \leqslant \sqrt{2n} \leqslant p + 1$, results in a competitive ratio for A bounded above by $1/(\sqrt{n/2} - 1)$. ▮

3 Relaxed models and charges

As we have already mentioned, we study in this section two further relaxations of the on-line model introduced in [3]. Recall that in this model:

- graph is revealed by clusters;
- an on-line algorithm builds its solution irrevocably choosing at each iteration which among vertices of the cluster just arrived will be included in the solution under construction.

Relaxations considered for this model are based upon weakening constraints of irrevocability. We will assume that algorithm can, during iteration i, includes in the solution also vertices revealed during iterations $j < i$. Such a relaxation can, of course, be very weak (permissive) since, if no additional assumption is made, algorithm can wait until the whole of graph is revealed before making any choice of the solution; in this case, on-line MAX INDEPENDENT SET-model becomes the classical off-line MAX INDEPENDENT SET. To avoid such situation, we introduce charges penalizing freedom: delayed choice of a vertex will be charged in such a way that its contribution in the final independent set will be smaller than 1; furthermore, the later a vertex chosen, the smaller its contribution in the final solution. More precisely, we will assume that if a vertex revealed during iteration j, is included in the solution during iteration $i \geqslant j$, then its real value in this solution is $1/k^{i-j}$(where $k > 1$ is a real number). Under this assumption, one can consider that, in iteration i, algorithm has to run on a vertex-weighted graph, where weights are as follows:

– vertices just arrived (i.e., arrived in iteration i) receive weight 1;
– vertices arrived in iteration $i - 1$ are weighted by $1/k$;
– ...
– vertices arrived in iteration 1 are weighted by $1/k^{i-1}$

Following this model, the real objective for an on-line algorithm A is to compute not really a maximum-size independent set but rather a maximum-weight one. The competitive ratio associated with this model is $\mathrm{val}_{\mathtt{A}}(S')/\alpha(G)$, where S' is the independent set computed by A and $\mathrm{val}_{\mathtt{A}}(S')$ its total weight.

In Section 3.1 we will assume that inclusion of a vertex in the solution under construction is irrevocable. Next, in Section 3.2, we further relax our model assuming that algorithm can backtrack, i.e., that it can even remove from current solution a vertex previously introduced. Note that charge-system makes that even this further relaxation remains interesting to be studied. Note finally that analogous charging-models can be assumed for the on-line model of Section 2. As it is shown in [6], results obtained are completely similar.

In what follows, we denote by t the number of steps needed that the final graph is fixed, by n the order of the final graph G, by G_i the ith cluster, by n_i the order of G_i and by H_i the part of G known at step i, i.e., the subgraph of G induced by vertices arrived during steps $1, \dots, i$ ($H_t = G$). If A is an on-line algorithm, then $\mathtt{A}(H_i)$ will denote the solution built by A up to ith iteration.

3.1 First irrevocability relaxation

As previously in Section 2.2, we will use a threshold algorithm. Such a use is due to the “blindness” of the algorithm entailed by the fact that choices are irrevocable, at least dealing with inclusions of vertices in the solution. This means that, once on-line algorithm A makes a choice, this choice can be fatal since the way the rest of the graph is revealed can forbid it from making any other extension of the solution it is constructing.

We have already mentioned that the charge-system we have considered, makes that the graph where A works can be assumed weighted as it has been described just previously. Consider then an off-line algorithm A solving (the off-line version of) WEIGHTED MAX INDEPENDENT SET and the following on-line algorithm, denoted by WOLTA, where threshold $r(n,t,k)$ will be precisely specified later:

– set $i = 1$;
– while $i \leqslant t$ and $\mathrm{val}(\mathtt{A}(H_i)) < r(n,t,k)$, set $i = i + 1$;
– if $i \leqslant t$ output $\mathtt{A}(H_i)$, else output a vertex of G_t.

Proposition 2 *If A achieves approximation ratio $\rho(n)$ for* WEIGHTED MAX INDEPENDENT SET*, then WOLTA achieves (under the model assumed) competitive ratio bounded below by*

$$\sqrt{\frac{\rho(n)}{n\left(t-\frac{t-1}{k}\right)}}$$

Proof. If WOLTA outputs a solution before iteration t of graph revealing, then it guarantees competitive ratio

$$\frac{\mathrm{val}(\mathtt{WOLTA}(G))}{\alpha(G)} \geqslant \frac{r(n,t,k)}{n} \tag{5}$$

Otherwise, for any iteration, $\mathrm{val}(\mathtt{WOLTA}(H_i)) \leqslant r(n,t,k)$. Since A guarantees approximation ratio $\rho(n)$,

$$\frac{\mathrm{val}\left(\mathtt{A}\left(H_i\right)\right)}{\alpha_w\left(H_i\right)} \geqslant \rho\left(\left|H_i\right|\right) \geqslant \rho(n) \tag{6}$$

where $\alpha_w(H_i)$ denotes the weighted stability number (i.e., the cardinality of a maximum-weight independent set) of H_i.

Let S^* be a maximum independent set of G ($|S^*| = \alpha(G)$) and consider integer sequence $a_i = |S^* \cap V(G_i)|$; obviously, $S^* \cap V(H_i)$ is an independent set of H_i, i.e., a feasible solution of WEIGHTED MAX INDEPENDENT SET on H_i. The value of this solution is at most the optimal one, i.e., $a_i + (a_{i-1}/k) + \ldots + (a_1/k^{i-1}) \leqslant \alpha_w(H_i)$. We so get, for all i:

$$\begin{aligned} r(n,t,k) \;&\geqslant\; \mathrm{val}\left(\mathtt{A}\left(H_i\right)\right) \;\geqslant\; \rho(n)\alpha_w\left(H_i\right) \\ &\qquad\qquad\qquad\quad\;\; \geqslant\; \rho(n)\left(a_i + \frac{a_{i-1}}{k} + \ldots + \frac{a_1}{k^{i-1}}\right) \\ a_i + \frac{a_{i-1}}{k} + \ldots + \frac{a_1}{k^{i-1}} \;&\leqslant\; \frac{r(n,t,k)}{\rho(n)} \end{aligned} \tag{7}$$

From inequalities of (7) we can show that:

$$\alpha(G) = a_t + a_{t-1} + \ldots + a_1 \leq \left(t - \frac{t-1}{k}\right) \frac{r(n,t,k)}{\rho(n)} \tag{8}$$

(the proof of (8) is given just after the end of the current proof). Using (6), we get from (8) a competitive ratio:

$$\frac{\mathrm{val}(\mathtt{WOLTA}(G))}{\alpha(G)} \geqslant \frac{\rho(n)}{\left(t - \frac{t-1}{k}\right) r(n,t,k)} \tag{9}$$

Ratio given by 5 is increasing with $r(n,t,k)$, while the one given by 9 is decreasing with $r(n,t,k)$. Equality of both ratios holds for

$$r(n,t,k) = \sqrt{\frac{n\rho(n)}{t - \frac{t-1}{k}}}$$

In this case, the competitive ratio achieved by WOLTA is as claimed. ▮

We now prove inequality in (8). Using (7), we will show that $\forall j \in \{0,1,\ldots,t-1\}$:

$$a_t + a_{t-1} + \ldots + a_{j+1} + \frac{a_j}{k} + \frac{a_{j-1}}{k^2} + \ldots + \frac{a_1}{k^j} \leqslant \left(t - j - \frac{t-j-1}{k}\right) \frac{r(n,t,k)}{\rho(n)} \tag{10}$$

Inequality in (10) is true for $j = t-1$; it is indeed inequality of (7) taking $i = t$. Suppose (10) true for $j > 0$; take also (7) for $i = j$ and multiply it by $1 - 1/k$; sum the result of the operation on (7) with (10). Then,

$$\begin{aligned} & a_t + a_{t-1} + \cdots + a_{i+1} + a_j\left(\frac{1}{k} + 1 - \frac{1}{k}\right) + \\ & + a_{j-1}\left(\frac{1}{k^2} + \left(1 - \frac{1}{k}\right)\frac{1}{k}\right) + \ldots + a_1\left(\frac{1}{k^j} + \left(1 - \frac{1}{k}\right)\frac{1}{k^{j-1}}\right) \\ \leqslant \;\; & \frac{r(n,t,k)}{\rho(n)}\left(t - j - \frac{t-j-1}{k} + 1 - \frac{1}{k}\right) \end{aligned}$$

that is exactly (10) in range $j-1$; this inequality is true for any j. We get (8) taking $j = 0$ and the proof is complete. Note that competitive ratio obtained in Proposition 2 is slightly better than ratio $\sqrt{\rho(n)/(nt)}$ obtained (without any relaxation) in [3]. However, both ratios remain of the same order.

Remark also that algorithm consisting of waiting until the whole of graph is revealed before running A on it, trivially guarantees competitive ratio $\rho(n)/k^{t-1}$ since vertex-weights are at least equal to $1/k^{t-1}$. Since n, k and t are known in advance, one has just, before running any algorithm, to compute values of $\rho(n)/k^{t-1}$ and of the ratio claimed by Proposition 2 and to run the algorithm associated to the best of these two values. Consequently, the following corollary holds and concludes the section.

Corollary 3 *If* A *is an off-line approximation algorithm for* WEIGHTED MAX INDEPENDENT SET, *achieving approximation ratio* $\rho(n)$, *then there exist an on-line algorithm for the model considered achieving competitive ratio at least*

$$\max\left\{\frac{\rho(n)}{k^{t-1}},\sqrt{\frac{\rho(n)}{n\left(t-\frac{t-1}{k}\right)}}\right\}$$

3.2 Further relaxation

In Section 3.1, we have relaxed irrevocability, allowing the algorithm to enter in the solution it constructs vertices arrived during previous iterations. In this section, we further can also remove from the current solution vertices entered it during former iterations. The charge-system considered here remains the same as in Section 3.1. In what follows, we still use notations introduced previously.

Consider the following algorithm, denoted by BOLA and using an off-line algorithm A solving WEIGHTED MAX INDEPENDENT SET:

- set $r = 0$;
- for $i = 1$ to t: if $r < val(\mathtt{A}(H_i))$, then set: $S = \mathtt{A}(H_i)$, $r = \mathrm{val}(\mathtt{A}(H_i))$ and $i = i+1$;
- output S.

In fact the work of BOLA amounts in determining an independent set for any H_i and in returning the best among them, i.e.,

$$\mathrm{val}(\mathtt{BOLA}(G)) = \max\left\{\frac{\mathrm{val}\left(\mathtt{A}\left(H_1\right)\right)}{\alpha(G)},\ldots,\frac{\mathrm{val}\left(\mathtt{A}\left(H_t\right)\right)}{\alpha(G)}\right\}$$

Let S^* be a maximum independent set of G. Set $a_i = |S^* \cap V(G_i)|$ and let b_i be the value of $S^* \cap V(H_i)$ in the weighted graph H_i; then, $b_i = a_i + (a_{i-1}/k) + (a_{i-2}/k^2) + \ldots + (a_1/k^{i-1})$.

Theorem 5 *If* A *achieves approximation ratio* $\rho(n)$ *for* WEIGHTED MAX INDEPENDENT SET, *then* BOLA *achieves competitive ratio* $\rho(n)/(t-((t-1)k))$.

Proof. For any $i \in \{1,2,\ldots,t\}$:

$$\frac{\mathrm{val}\left(\mathtt{A}\left(H_i\right)\right)}{\alpha_w\left(H_i\right)} \geqslant \rho\left(|H_i|\right) \geqslant \rho(n) \tag{11}$$

Obviously, $S^* \cap V(H_i)$ is a feasible solution of WEIGHTED MAX INDEPENDENT SET on H_i; so:

$$\alpha_w\left(H_i\right) \geqslant \mathrm{val}\left(S^* \cap V\left(H_i\right)\right) = b_i \tag{12}$$

Using (11) and (12) and the way final solution is built by BOLA, we get:

$$\mathrm{val}(\mathtt{BOLA}(G)) \geqslant \mathrm{val}\left(\mathtt{A}\left(H_i\right)\right) \geqslant \rho(n)b_i = \rho(n)\left(a_i + \frac{a_{i-1}}{k} + \ldots + \frac{a_1}{k^{i-1}}\right) \tag{13}$$

Using (13) and the same arguments as for the proof of inequality in (8), one immediately reaches:

$$\left(t - \frac{t-1}{k}\right)\mathrm{val}(\mathtt{BOLA}(G)) \geqslant \rho(n)\left(a_1 + a_2 + \ldots + a_t\right) = \rho(n)\alpha(G)$$

that directly deduces the result claimed. ∎

From Theorem 5, one easily sees that relaxation admitted in this section improves largely result of Proposition 2. For instance, if t is a fixed constant, BOLA reaches, despite of charges, competitive ratio of the same order as the approximation ratio of the off-line algorithm A that uses as a sub-routine.

Proposition 3 *If $n \geqslant t(t+1)/2$, then no on-line algorithm can achieve competitive ratio (for the model dealt) better than $1/t(1-(1/k))$.*

Proof. Let A be any on-line algorithm. Consider the following way of revealing G:

- in first iteration, one reveals a clique on t vertices, in the second one a clique of size $t-1$ and so on until the $(t-1)$th iteration where a clique of size 2 is revealed; the tth cluster will be a clique on the $n-(t(t-1)/2)$ remaining vertices;
- in iteration i, we link vertices of $\mathtt{A}(H_{i-1})$ (i.e., the ones chosen by A in iteration $i-1$) to all of the vertices of the clique revealed in step i.

We show that, in any iteration, $\mathrm{val}(\mathtt{A}(H_i)) \leqslant 1 + 1/k + \ldots + 1/k^{i-1}$. Indeed, $\mathtt{A}(H_i)$ contains at most one vertex in any cluster since these clusters are cliques. Furthermore, upon the arrival of cluster i any vertex in $\mathtt{A}(H_{i-1})$ is linked with $V(G_i)$. So, if A chooses a vertex in $V(G_i)$, then $\mathtt{A}(H_{i-1}) \cap \mathtt{A}(H_i) = \emptyset$. Since vertices of jth cluster have weight $1/k^{i-j}$, we deduce the relation claimed. So,

$$\mathrm{val}(\mathtt{A}(G)) \leqslant 1 + \frac{1}{k} + \ldots + \frac{1}{k^{t-1}} \leqslant \frac{1}{1-\frac{1}{k}} \tag{14}$$

Now, it suffices to note that

$$\alpha(G) = t \tag{15}$$

Indeed, cluster (clique) arrived in iteration i has size $t+1-i$. and we link at most one vertex of cluster i to any vertex of clusters $j > i$. So, in any cluster, there exists at least a vertex v not linked to vertices of the subsequent clusters. The set of all these vertices v forms an independent set of cardinality t and (15) is true. Combining (14) and (15) we get: $\mathrm{val}(\mathtt{A}(G))/\alpha(G) \leqslant 1/(t(1-(1/k)))$, qed. ∎

From Theorem 5 and Proposition 3, one can see that BOLA, although simple is quite competitive since, considering an optimal off-line algorithm instead of A, its competitive ratio becomes $1/(t(1-(1/k))+1/k)$ that is very close to upper bound given by Proposition 3.

4 Conclusion

We have presented new models for on-line MAX INDEPENDENT SET. In addition to results themselves, methods used are interesting per se since they exhibit how on-line computation can be seen as extension of polynomial approximation theory. In particular,

- algorithms devised are, for the most of them, very competitive, since their competitive ratios match upper bounds provided for the models dealt;
- competitive analyses take advantage of existing approximation results and hence, they can be seen as reductions from approximation to on-line framework.

Two major open directions that studies as the ones of the paper address are: first, the development of opposite-sense reductions, i.e. reductions from the on-line to the approximation framework and, second, development of reductions between on-line models for the same or, mainly, for distinct combinatorial optimization problems.

Références

[1] G. Ausiello, E. Feuerstein, S. Leonardi, L. Stougie, and M. Talamo. Algorithms for the on-line traveling salesman problem. *Algorithmica*, 29(4):560–581, 2001.

[2] C. Berge. *Graphs and hypergraphs*. North Holland, Amsterdam, 1973.

[3] M. Demange, X. Paradon, and V. Th. Paschos. On-line maximum-order induced hereditary subgraph problems. In V. Hlaváč, K. G. Jeffery, and J. Wiedermann, editors, *SOFSEM 2000—Theory and Practice of Informatics*, volume 1963 of *Lecture Notes in Computer Science*, pages 326–334. Springer-Verlag, 2000.

[4] M. Demange and V. Th. Paschos. Improved approximations for maximum independent set via approximation chains. *Appl. Math. Lett.*, 10(3):105–110, 1997.

[5] M. Demange and V. Th. Paschos. Improved approximations for weighted and unweighted graph problems. *Theory of Computing Systems*, 2004. To appear.

[6] B. Escoffier. Problème *on-line* du stable de cardinalité maximale. Mémoire de DEA, 2002.

[7] M. M. Halldórsson. Approximations via partitioning. JAIST Research Report IS-RR-95-0003F, Japan Advanced Institute of Science and Technology, Japan, 1995.

[8] M. M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. *J. Graph Algorithms Appli.*, 4(1):1–16, 2000.

[9] M. M. Halldórsson, K. Iwama, S. Miyazaki, and S. Taketomi. Online independent sets. *Theoret. Comput. Sci.*, 289(2):953–962, 2002.

[10] D. S. Hochbaum, editor. *Approximation algorithms for NP-hard problems*. PWS, Boston, 1997.

[11] V. Th. Paschos. A survey about how optimal solutions to some covering and packing problems can be approximated. *ACM Comput. Surveys*, 29(2):171–209, 1997.

Modélisation du problème général d'ordonnancement de véhicules sur une ligne de production et d'assemblage

Vincent **Giard**[1] & Jully **Jeunet**[2]

Résumé

Plusieurs formulations partielles du problème d'ordonnancement sur lignes de production ou d'assemblage dédiées à une production de masse fortement diversifiée – l'industrie automobile constituant l'exemple le plus cité – ont été proposées. Des hypothèses implicites limitent la portée de beaucoup d'entre elles. On proposera ici une formulation générale du problème d'ordonnancement s'appuyant sur des hypothèses réalistes de description du processus physique et prenant en compte l'incidence économique des décisions d'ordonnancement, notamment au niveau de l'appel momentané de renforts et de la prise en compte de l'incidence de rafales sur certains coûts de lancement (atelier de peinture, par exemple). Le modèle linéaire obtenu permet une description explicite et complète de ce problème complexe. La résolution de problèmes réels par la programmation mathématique est difficile en raison de la taille du problème. La description proposée facilite la mise au point d'heuristiques pertinentes.

Mots clefs : ligne de production et d'assemblage, ordonnancement sur ligne de produits hétérogènes, contraintes d'espacement, rafales, personnel de renfort, évaluation économique.

Abstract

Several formulations to the mixed-model assembly line have been proposed in the literature. Such a problem is concerned with the production of a variety of product models. Vehicles fall into that category. The underlying assumptions of those models are often restrictive, therefore limiting their applicability. Our incentive is to develop a general formulation of the car sequencing problem that explicitly takes into account the constraints of the paint shop, those of the body shop and finally the constraints emanating from the assembly shop. The resultant linear program captures the whole complexity of the problem as it is encountered in practice. Solving real-sized instances to optimality would require too much computation time. The proposed description provides a starting point to the development of relevant heuristic solution approaches.

Key Words: mixed model assembly line, car sequencing, spacing or contiguity constraints, colour grouping constraints, additional workers, economic evaluation.

[1] Lamsade, université Paris Dauphine, {giard@lamsade.dauphine.fr}.

[2] CNRS, Lamsade, université Paris Dauphine, {jeunet@lamsade.dauphine.fr}.

1 Introduction

Au début du siècle dernier, les consommateurs étaient largement contraints dans leurs choix de véhicules par l'offre disponible, comme l'illustre la célèbre déclaration d'Henri Ford : le client peut tout choisir, même la couleur, à condition qu'elle soit noire. Les temps ont changé et l'espace de choix en matières d'options n'a désormais plus de limite. Ce problème est maintenant général, la tendance étant de produire en masse des produits personnalisés mais c'est dans l'industrie automobile que la complexité est maximale, ce qui explique que ce secteur soit pris à titre d'exemple. La diversité est obtenue par le montage de modules alternatifs (moteurs de différentes puissances, par exemple) et d'options demandées seulement par certains clients (toit ouvrant, par exemple). Se pose alors le problème du séquencement des véhicules sur la ligne, sachant que la variété conduit, sur certains postes, à des temps opératoires pouvant être supérieurs ou inférieurs au temps de cycle. On est alors confronté au problème qualifié d'***ordonnancement sur ligne de produits hétérogènes*** qui cherche à éviter des phénomènes de désamorçage ou de saturation conduisant à des arrêts de la ligne.

À cet objectif de lissage de la charge de travail, s'ajoute celui du lissage de la consommation de composants non montés systématiquement, en particulier lorsque ceux-ci sont approvisionnés par kanban. La littérature consacrée au traitement de ces deux objectifs est abondante et certaines contributions considèrent conjointement les deux objectifs dans des modèles d'optimisation multicritères.

À ce stade, on doit se rappeler que les usines de production automobile se composent généralement de trois lignes successives sur lesquels les véhicules se succèdent normalement dans le même ordre : la tôlerie, peinture et assemblage. Chaque ligne possède ses propres contraintes. Sur la ligne de peinture, il est économiquement souhaitable d'obtenir un ordonnancement regroupant les véhicules par rafale de teinte, tandis que sur les lignes d'assemblage le respect des contraintes d'espacement permet l'équilibrage de la charge. Ces objectifs ne sont pas faciles à concilier et, dans la littérature, ces contraintes de regroupement des couleurs sont ignorées à quelques rares exceptions près (Giard, 1997, [5] et 2003, [6]). Cet article est sans doute l'un des premiers à considérer simultanément toutes les contraintes dans une formulation optimale du problème de séquencement.

Le § 2 est consacré à une revue des modèles de la littérature relative au problème de l'ordonnancement de produits hétérogènes sur une ligne d'assemblage. Ces modèles sont groupés en trois catégories, en fonction de l'objectif de minimisation poursuivi. Le § 3, page 241 propose une formulation générale du problème d'ordonnancement des véhicules, tel qu'il se rencontre en pratique. Au lieu de modéliser successivement les contraintes provenant des trois ateliers (tôlerie, assemblage et peinture) nous adoptons une différenciation des contraintes fondées sur l'élasticité ou non de la capacité des postes de travail d'où ces contraintes sont issues. Ainsi, les postes capacitaires sont ceux dont la capacité est résolument fixe. La majorité des postes de tôlerie en font partie. Les postes non-capacitaires peuvent voir leur capacité s'accroître en accueillant des travailleurs supplémentaires ou renforts. Nous distinguons un troisième groupe de contraintes relatives à l'atelier de peinture. Le § 3, page 241 conclut en rappelant les apports de la modélisation proposée pour la création de méthodes de résolution exacte et heuristique.

2 Revue de la littérature

Comme nous l'avons souligné en introduction, deux objectifs principaux sont poursuivis dans le problème de séquencement : lisser la charge de travail (§ 2.1), maintenir la consommation de pièces aussi constante que possible (§ 2.2, page 238). Dans les systèmes juste-à-temps, le deuxième objectif est primordial tandis que le premier est secondaire. Certains auteurs considèrent ces deux objectifs comme d'égale importance (§ 2.3, page 239). On terminera par une présentation synthétique des hypothèses implicitement retenues (§ 2.4, page 240), ce qui conduira à la formulation générale que nous proposons.

2.1 Lissage de la charge

Cet objectif équivaut également à minimiser les arrêts du convoyeur. L'une de ses formulations est connue sous le nom de problème de la variation du taux de produit. Soit u_i le nombre d'unités du modèle i à séquencer, avec $i = 1,\ldots$, M et soit N, le nombre total de véhicules. On a $\mathrm{N} = \sum_{i=1}^{\mathrm{M}} u_i$.

Soit x_{ij} le nombre d'unités de modèle i séquencé entre les positions 1 et j. Le taux idéal de produit i dans la séquence de véhicules est $r_i = u_i/\mathrm{N}$. Entre les positions 1 et j, le nombre idéal de modèles i est égal à jr_i, son nombre réel est x_{ij}. L'objectif est donc de minimiser la distance entre le nombre idéal et le nombre réel de chacun des modèles. Ceci s'écrit : $\sum_{j=1}^{\mathrm{N}} \sum_{i=1}^{\mathrm{M}} (x_{ij} - jr_i)^2$.

Une autre façon d'aborder le problème est d'imposer des contraintes d'espacement entre modèles identiques de sorte à réguler le rythme de la production. Ces contraintes peuvent être interprétées comme des contraintes d'équilibrage de la charge mais ces formulations ne sont pas nécessairement équivalentes (voir remarque 1 du § 3.2.1.3, page 244).

Solution optimale. Mitsumori ([15], 1969) propose une méthode de résolution optimale pour minimiser le risque d'arrêt du convoyeur. Miltenburg et *al* ([14], 1990) fournissent une formulation du problème par la programmation dynamique qui peut être résolue facilement pour des problèmes de petite taille. Kubiak et Sethi (1991, [11]; 1994, [12]) montrent que le problème peut être formulé comme un problème d'assignation et résolu à l'aide d'algorithmes ad hoc. Xiaobo et Ohno (1994) développent un algorithme de séparation-évaluation (*branch and bound*) permettant de trouver la solution optimale pour des petites instances. Récemment, Bautista et *al*. ([1], 2000) ont montré qu'une solution optimale au problème consiste en la répétition d'une sous-séquence optimale. Plus généralement, les auteurs montrent que cette propriété est valable pour n'importe quelle fonction objectif $\sum_{j=1}^{N} \sum_{i=1}^{M} \phi(x_{ij} - jr_i)$ où ϕ est une fonction convexe positive possédant un minimum global en 0. Lorsque ces conditions sont remplies, la solution du problème originel peut être déduite de la solution au problème réduit.

Notons que ces méthodes optimales sont difficilement appropriées pour la résolution de cas réels (entre 500 et 1500 véhicules) pour des raisons de temps de calcul.

Approches heuristiques. Ding et Cheng ([4], 1993) proposent une procédure qui minimise l'écart entre le nombre d'unités produites jusqu'à présent et les quantités idéales pour la période (étape) courante et la période suivante. Korkmazel et Meral ([10], 2001) modifient cette procédure de sorte à considérer davantage de véhicule-candidat dans un éventail plus large de priorités. Les résultats expérimentaux (jusqu'à 15 modèles) montrent la supériorité de cette procédure modifiée. Choi et Shin ([3], 1997) présentent une simple règle de décision pour sélectionner à chaque étape (ou période) le véhicule qui minimise pour chaque option l'écart entre l'espacement courant et la limite spécifiée de cet espacement. Gottlieb et *al.* ([7], 2003) décrivent plusieurs heuristiques gloutonnes et montrent que leurs versions dynamiques sont plus performantes que leurs versions statiques. Ces heuristiques consistent à choisir de manière itérative le véhicule qui minimise le nombre de violations de contraintes tout en nécessitant le nombre maximum d'options.

Pour des problèmes de taille moyenne (jusqu'à 90 véhicules), Xiaobo et Ohno ([21], 1997) mettent en œuvre avec succès un algorithme de recuit simulé. Cet algorithme n'incorpore pas d'éléments spécifiques au problème traité puisqu'une seule règle de transition est utilisée; cette dernière consistant à échanger deux véhicules dans la séquence. Gottlieb et *al*. ([7], 2003) développent une métaheuristique de type optimisation par colonies de fourmis. Le problème est modélisé comme un graphe dans lequel chaque sommet représente un véhicule et chaque arc (*i,j*) porte un poids qui mesure la désirabilité du séquencement de *j* après *i*. Une stratégie élitiste est utilisée pour sélectionner le meilleur véhicule. La performance de cette heuristique est comparée à celle d'un algorithme de seuil (*threshold algorithm*) utilisant six règles de transition pour la génération de voisinages. Smith et *al*. ([18], 1996) considèrent explicitement le problème de séquencement avec satisfaction des contraintes d'espacement. Les violations sont prises en compte dans la fonction objectif par l'utilisation d'une matrice de proximité indiquant le niveau de pénalité pour tout couple de véhicules identiques violant la contrainte d'espacement. Un algorithme de descente est appliqué et se retrouve systématiquement coincé dans des optima locaux. Un algorithme de recuit simulé est alors utilisé. Enfin, les approches de type réseaux de neurones d'Hopfield sont adoptées. Les résultats des simulations montrent la supériorité du recuit simulé sur l'algorithme de descente. Les réseaux neuronaux permettent l'obtention de meilleures solutions que celles obtenues par le recuit, à mesure que la taille du problème augmente.

2.2 Maintien à un niveau constant de la consommation de pièces

Cet objectif est celui initialement poursuivi par Toyota dans un environnement de juste-à-temps.

Solution optimale. Bautista et *al*. ([2], 1996) montrent que le problème est équivalent à la recherche d'un chemin minimum dans un graphe de sorte que n'importe quel algorithme de détermination de chemin extrême peut être utilisé pour obtenir une solution optimale au problème. Les auteurs reconnaissent néanmoins l'inaptitude de ces algorithmes à déterminer la solution optimale pour des problèmles de taille réelle en raison du nombre trop important d'arcs.

Heuristiques. Parmi elles se trouve l'heuristique de Monden, connue sous le nom de *Goal Chasing Method* ([16], 1983 ou 1998). Sumichrast et *al*. ([19], 1992) évaluent cette méthode (GCM I) et sa version simplifiée (GCM II) développée chez Toyota. Les auteurs proposent une méthode fondée sur une idée similaire et incluent dans leur expérience de simulation un algorithme proposé par Miltenburg ([14], 1989). Quatre mesures de l'inefficacité de la chaîne sont utilisées: l'incomplétude du travail (le ratio du travail inachevé sur le travail total), l'inactivité (pourcentage du temps durant lequel les travailleurs sont oisifs), le temps passé au poste de travail (pourcentage de temps qu'un travailleur passe à son propre poste de travail; les travailleurs étant autorisés à quitter leur poste pour fournir de l'aide aux postes en congestion), la variance des taux d'utilisation des composants. La procédure développée par les auteurs utilise la même méthode de sélection des véhicules que celle employée dans GCM mais la fonction-objectif est fondée sur la durée totale d'assemblage à tous les postes de travail. Les résultats des simulations montrent la bonne performance de cette procédure. Leu et *al*. ([13], 1996) développent un algorithme génétique pour minimiser la variabilité de la consommation de pièces. Les auteurs montrent la supériorité de leur algorithme sur l'heuristiquede Monden dans la plupart des cas examinés.

2.3 Optimisation conjointe des deux objectifs précédents

Solution optimale. Korkmazel et Meral ([10], 2000) traitent le problème de séquencement bi-critère et proposent une formulation en un problème d'assignation (consécutivement aux travaux de Kubiak et Sethi, [12], 1994) qui peut être résolu optimalement pour des cas de petites tailles (jusqu'à 15 véhicules).

Heuristiques. Hyun et *al*. ([8], 1998) conçoivent un algorithme génétique multi-objectif. Ils considèrent trois objectifs d'égale importance: minimiser l'utilisation de renforts pour faire face aux surcharges de travail, minimiser le coût de lancement, maintenir constant le taux d'utilisation des pièces. La réalisation simultanée de ces trois objectifs n'est pas une tâche évidente dans la mesure où chacun des objectifs peut être conflictuel avec les autres. Les solutions de ces problèmes multi-critères sont des solutions de type optima de Pareto dans lesquelles il est par définition impossible d'améliorer un objectif sans détériorer les autres. L'approche classique de ces problèmes consiste à réduire les différents objectifs en une fonction unique dans laquelle chaque critère occupe un certain poids. Ceci pose le problème de l'affectation des poids ou de manière équivalente, le problème de la définition de l'importance relative de chaque objectif. Les auteurs développent un processus de sélection qui se passe de toute décision subjective. Le processus de sélection fonctionne de la manière suivante. Une solution localisée dans un cube de faible densité possède une plus grande probabilité de survie que toute autre solution située dans un cube de plus forte densité inclus dans la même strate parétienne.

L'algorithme génétique qui en résulte fournit de meilleures solutions que ceux produits par les algorithmes génétiques multi-objectifs traditionnels.

Murata et *al*. ([17], 996) proposent une méthode d'affectation de poids variables aux différentes fonctions objectif de sorte à rendre la recherche variable. L'idée est d'exploiter plusieurs directions dans la recherche de solutions pareto-optimales.

Tamura et Ohno ([20], 1999) considèrent le problème bi-critère consistant à lisser le taux d'utilisation des composants et la charge de travail à chaque station d'une ligne comportant une ligne principale ainsi qu'une sous-ligne ou brin supplémentaire. L'installation de ce brin se justifie par l'existence de produits nécessitant de longs temps d'assemblage. Trois méthodes de résolution sont proposées et testées sur un nombre limité de cas. La méthode *Goal Chasing* est adaptée à ce cas de ligne comportant une sous-ligne, un algorithme de recherche «tabou» est mis en œuvre et une approche par la programmation dynamique est développée pour l'obtention de solutions optimales. Les solutions optimales ne peuvent être obtenues que pour des problèmes de petite taille. La méthode «tabou» fournit des solutions satisfaisantes tandis que la méthode *Goal Chasing* apparaît comme l'algorithme le plus rapide.

2.4 Hypothèses implicites des approches retenues

Les différentes approches étudiées peuvent s'analyser à travers plusieurs grilles permettant de mieux cerner la classe de problèmes traités. La première est relative au lissage de la consommation des composants. La seconde concerne la prise en compte possible du coût de lancement lié au changement d'une caractéristique de deux véhicules se suivant dans l'ordonnancement, par exemple un changement de teinte de peinture; on parlera alors de problème de rafales. La troisième est liée à l'incidence de la variabilité de certains temps opératoires, en fonction de caractéristiques des véhicules. Ce point est à l'origine des principales difficultés de l'ordonnancement. Pour des raisons qui seront explicitées, on est amené à distinguer selon que n'est affecté ou non qu'un seul poste de la ligne. Si plusieurs postes sont concernés, la capacité de ces postes peut être intangible – on parlera alors de postes capacitaires – ou non; dans ce dernier cas, une surcharge momentanée de travail peut être absorbée par des renforts.

Modélisation implicite	Prise en compte du lissage de consommation de composants	Prise en compte de problèmes de rafales	Prise en compte de la variabilité de temps opératoires sur certains postes		
			Postes non capacitaire		Postes capacitaires
			Un seul poste concerné	Plusieurs postes concernés	Plusieurs postes concernés
§ 2.1	Non	Non	Oui (implicitement[1])	Non	Non
§ 2.2	Oui	Non	Non	Non	Non
§ 3	Non	Oui	Oui	Oui	Oui

[1] Voir remarque 1 du § 3.2.1.3, page 244.

3 Formulation du problème d'ordonnancement

3.1 Ensemble à ordonnancer

Soit un ensemble de N véhicules à ordonnancer. On note x_{ij} une variable binaire valant 1 si le véhicule i ($i = 1, \ldots,$N) a le rang j.

Un seul des véhicules i peut avoir le rang j, ce que traduit la relation 1 (N contraintes) :

$$\sum_{i=1}^{N} x_{ij} = 1 \text{, pour } j = 1, \ldots, N \qquad \textit{relation 1}$$

Le véhicule i se voit attribuer nécessairement un seul des rangs j, ce que traduit la relation 2 (N contraintes) :

$$\sum_{j=1}^{N} x_{ij} = 1 \text{, pour } i = 1, \ldots, N \qquad \textit{relation 2}$$

3.2 Contraintes relatives à la variabilité de la charge de travail sur certains postes

La production de masse de produits personnalisés conduit à une différenciation obtenue par un ensemble d'options (toit ouvrant…) pouvant être choisies ou non par le client et l'usage de jeux de modules assurant une même fonction souvent avec des performances techniques et économiques différentes (moteurs…) mais pas toujours (volant à gauche ou à droite, en fonction du client), un véhicule retenant nécessairement un module pour chacun des jeux. Cette personnalisation conduit à une variation de la charge de travail sur certains postes. L'origine de cette variation pouvant être liée à une option ou un module, on parlera de ***critère*** (mais les exemples choisis seront liés aux options). Celle-ci est prise en compte lors de la conception de la ligne sur la base de caractéristiques prévisionnelles moyennes (20% de toits ouvrants, par exemple) et conduit à affecter à ce type de poste une zone plus longue sur le convoyeur ou à prévoir un stock-tampon en aval du poste si celui-ci est fixe. Ces dispositions permettent de synchroniser les postes de la ligne en garantissant que les arrivées s'effectueront bien dans chaque poste de travail à la cadence définie pour la ligne. Il faut alors que l'ordonnancement respecte certaines contraintes que nous allons examiner, faute de quoi le processus sera perturbé par des mécanismes de saturation bloquant progressivement les postes de travail en amont (propagation-amont).

L'analyse du problème conduit à distinguer deux cas de figure. Certains de ces postes affectés par cette variabilité de travail impliquent le respect strict de ces contraintes, généralement en raison de la capacité d'équipements (robots de soudure…) ; on parle alors de ***postes capacitaires*** (qui se trouvent principalement en tôlerie). D'autres postes permettent de s'affranchir de cette contrainte à condition de renforcer le personnel affecté à ce poste, l'outillage utilisé pouvant sans problème s'adapter à cette variation du nombre d'opérateurs ; on parle alors de ***postes non capacitaires***. On commencera par analyser le cas des postes capacitaires (§ 3-2.1). L'analyse des postes non capacitaires en découle immédiatement et conduit à une transformation de contraintes et une modification de la fonction-objectif qui doit prendre en considération le coût additionnel induit par l'éventuelle mobilisation d'opérateurs supplémentaires en raison de l'ordonnancement retenu (§ 3-2.2, page 248).

3-2.1 Prise en compte des contraintes des postes capacitaires

3.2.1.1 Formulation initiale

On note *k* l'un des postes capacitaires de la ligne (sous-ensemble $\mathscr{K}_1$ de l'ensemble des postes de la ligne). La durée du travail requis par le véhicule *i* sur le poste de travail *k* est notée θ_{ik} et le temps de cycle est noté $\bar{\theta}$, l'arrivée des véhicules sur ce poste étant cadencée par ce temps de cycle.

On note *h*, le rang du véhicule positionné sur le poste de travail *k* (avec *h* = 1, …, N), le temps de travail à exécuter sur ce véhicule de rang *h* est $\sum_{i=1}^{N} \theta_{ik} x_{ih}$.

L'écart entre cette durée et le temps de cycle $\sum_{i=1}^{N} \theta_{ik} x_{ih} - \bar{\theta}$ correspond à un dépassement s'il est positif et, dans le cas contraire, à une éventuelle possibilité de rattrapage d'une surcharge de travail antérieure.

On note R$_{kh}$ l'excédent de charge de travail à résorber après traitement du véhicule de rang *h* $(R_{k0} = 0)$. Ce report est limité par une quantité de travail R_k^{max}, liée au déplacement maximum du véhicule sur le convoyeur ou à la taille des stocks-tampons situés en amont et aval du poste de travail *k* si le véhicule est traité par un poste fixe[1]. Le fait que ce report maximal R_k^{max} soit fixe est lié à l'hypothèse implicite qu'aucune compensation sur les temps opératoires n'est considérée comme possible avec les postes adjacents[2].

Ce report R$_{kh}$, qui ne peut être négatif, intègre la charge de travail non résorbée après traitement du véhicule de rang *h* – 1. Une fois traité le véhicule de rang *h,* cet excédent est: $R_{kh} = Max\left\{0, Min\left[R_{k,h-1} + \sum_{i=1}^{N} \theta_{ik} x_{ih} - \bar{\theta}, R_k^{max}\right]\right\}$. Cette contrainte, utilisée de manière récurrente à partir du premier véhicule ordonnancé sur le poste de travail *k*, traduit simplement le fait que la charge cumulée de travail exécuté ne peut dépasser la capacité cumulée de production en tenant compte des contraintes de report maximal. Sa transcription dans une formulation linéaire du problème se fait sans difficulté par l'intermédiaire des relations 3, en utilisant la variable intermédiaire W_{kh}.

[1] Le stock-amont permet d'attendre la libération du poste fixe et donc d'éviter la saturation. Dans le cas d'un poste embarqué sur convoyeur, ce stock est inutile, l'opérateur se déplaçant. Le stock-aval permet au poste aval de respecter la cadence du cycle et donc évite le désamorçage. Le dimensionnement de ces stocks correspond à l'arrondi supérieur du quotient de l'excédent maximal de travail à résorber (voir ci-après) par le temps de cycle.

[2] Dans ce cas, on peut se ramener au cas général en travaillant sur un poste fictif réunissant les postes concernés.

$$W_{kh} \le R_k^{max}\ ;\ W_{kh} \le R_{k,\,h-1} + \sum_{i=1}^{\mathrm{N}} \theta_{ik}x_{ih}\ ;\ R_{kh} \ge 0\ ;\ R_{kh} \ge W_{kh}\ \textit{pour } h = 1, \ldots, \mathrm{N}\ \textit{et}$$

$$k \in \mathcal{K}_1 \qquad\qquad \textit{relations 3}$$

Cette formulation comporte $4 \cdot \mathrm{N}$ contraintes pour chaque poste $k \in \mathcal{K}_1$, ce qui fait que l'on a intérêt à traiter des problèmes d'ordonnancement où la taille N de l'ensemble de véhicules à ordonnancer et le nombre $\mathcal{K}_1$ de postes critiques ne sont pas trop conséquents. Si plusieurs postes ont des contraintes «voisines», il suffit alors de ne s'intéresser qu'au plus pénalisant, les autres contraintes étant alors satisfaites. Ce nombre élevé de contraintes pour un poste critique s'explique par la très grande variété postulée de temps opératoires sur ce poste laquelle oblige à vérifier que le temps opératoire de chacun des véhicules ne compromet pas un éventuel rattrapage de retards imputables à un ou plusieurs véhicules de rang inférieur.

3.2.1.2 Formulation alternative par les contraintes d'espacement

Le nombre de contraintes peut diminuer très fortement si le nombre de temps opératoires se limite à 2 que l'on notera T_k^{max} (présence du critère A, toit ouvrant, par exemple) et T_k^{min} (absence du critère A, noté $\overline{\mathrm{A}}$).

Supposons que le premier véhicule ordonnancé ($h = 1$) possède le critère A ($\theta_{ik} = \mathrm{T}_k^{max}$). Lors de la conception de la ligne on a défini le report maximal R_k^{max} comme devant être supérieur ou égal au report $R_{k,\,1} = \mathrm{T}_k^{max} - \bar{\theta}$ enregistré après avoir ordonnancé ce premier véhicule.

Le véhicule ordonnancé suivant ne peut posséder le critère A que si le temps résiduel $R_k^{max} - R_{k,\,1} = R_k^{max} - (\mathrm{T}_k^{max} - \bar{\theta})$ est supérieur à l'accroissement de report $(\mathrm{T}_k^{max} - \bar{\theta})$. Le nombre maximal ν_k de véhicules consécutifs «consommant» ce report maximal R_k^{max} est donc l'entier inférieur du quotient $R_k^{max}/(\mathrm{T}_k^{max} - \bar{\theta})$. À la conception de la chaîne, on a intérêt donc à définir R_k^{max} comme un multiple de $(\mathrm{T}_k^{max} - \bar{\theta})$, ce que nous supposerons par la suite sans perte de généralité. Si les ν_k premiers véhicules possèdent le critère A, le véhicule suivant, de rang $h = \nu_k + 1$, ne peut pas posséder ce critère sans violer la contrainte du report maximum. Ce véhicule de rang $\nu_k + 1$ permet de diminuer le retard cumulé $R_{k,\,\nu_k} = R_k^{max}$ après le placement du véhicule de rang ν_k, d'un temps égal à $(\bar{\theta} - \mathrm{T}_{k_{min}})$. Sauf si l'excédent de travail $(\mathrm{T}_k^{max} - \bar{\theta})$ du temps opératoire d'un véhicule avec option par rapport au temps de cycle est inférieur au temps inutilisé $(\bar{\theta} - \mathrm{T}_k^{min})$ par un véhicule sans option, il faudra plusieurs véhicules sans option avant de pouvoir replacer un véhicule avec option. Ce nombre de véhicules μ_k est égal à la partie entière inférieure du quotient $(\mathrm{T}_k^{max} - \bar{\theta})/(\bar{\theta} - \mathrm{T}_k^{min})$. Il s'ensuit que ν_k véhicules consécutifs ayant le critère A sont nécessairement suivis de μ_k véhicules n'ayant pas ce critère. Mais, sur les $\nu_k + \mu_k$ premiers véhicules consécutifs, les ν_k véhicules ayant le critère A peuvent ne pas être consécutifs parce qu'il est évident qu'une telle situation conduit nécessairement le report $R_{k,\,\nu_k + \mu_k}$ du véhicule de rang $\nu_k + \mu_k$ à être inférieur à R_k^{max}. Dès lors, si la condition «**au plus** ν_k **véhicules avec critère sur** $\nu_k + \mu_k$ **véhicules consécutifs**» est respectée, les contraintes des relations 3 sont respectées.

Cette démonstration permet de remplacer, pour chaque poste $k \in \mathscr{K}_1$, les $4 \cdot \mathrm{N}$ contraintes des relations 3 par les $[\mathrm{N} - (\nu_k + \mu_k) + 1]$ contraintes des relations 4, qualifiées de **contraintes d'espacement**. Celles-ci jouent sur des fenêtres glissantes de temps (ou de séquences de véhicules, ce qui revient au même) bornées supérieurement par *h* et inférieurement par $h - (\nu_k + \mu_k) + 1$. Le temps de traitement du véhicule *h* sur le poste *k* $\sum_{i=1}^{\mathrm{N}} \theta_{ik} x_{ih}$ est égal à T_k^{min} ou à T_k^{max} mais, dans la transformation de la formulation, on ne s'intéresse plus qu'au fait que les véhicules séquencés possèdent ou non le critère A. Cette information est donnée par δ_{ik} qui vaut 1 si le véhicule *i* possède le critère A et 0, dans le cas contraire. D'une manière générale, un ensemble de N véhicules caractérisables par G critères est décrit par une matrice de booléens de terme général δ_{ig}, avec $i = 1,\ldots,\mathrm{N}$ et $g = 1,\ldots,\mathrm{G}$. Ne s'intéressant ici qu'au critère A, le nombre de véhicules possédant ce critère sur la fenêtre glissante de $(\nu_k + \mu_k)$ véhicules est $\sum_{j=\{h-(\nu_k+\mu_k)+1\}}^{h} \sum_{i=1}^{\mathrm{N}} \delta_{ik} \cdot x_{ij}$. Comme on l'a vu, ce nombre doit être inférieur à ν_k, ce qui conduit aux relations 4:

$$\sum_{j=\{h-(\nu_k+\mu_k)+1\}}^{h} \sum_{i=1}^{\mathrm{N}} \delta_{ik} \cdot x_{ij} \le \nu_k,\ pour\ h = \nu_k + \mu_k, \ldots, \mathrm{N}\ et\ k \in \mathscr{K}_1\ avec$$

$$\mu_k = (\mathrm{T}_k^{max} - \bar{\theta})/(\bar{\theta} - \mathrm{T}_k^{min})\ et\ \nu_k = R_k^{max}/(\mathrm{T}_k^{max} - \bar{\theta}) \qquad relations\ 4$$

Ces contraintes d'espacement, utilisées depuis un certain temps dans l'industrie automobile, ont été introduites dans la littérature depuis quelques années (Smith et *al.*, [18], 1996, Giard, [5], 1997; Choi et *al.*, [3], 1997). Leurs justifications et limites par l'analyse fine du processus de production n'ont jamais été apportées à notre connaissance.

3.2.1.3 Remarques

Six remarques peuvent être faites.

- **Lissage**. Si un seul poste est concerné par cette variabilité, on se retrouve implicitement dans le cas d'un problème de lissage, si la contrainte d'espacement se rapproche de 1/N. Dans le cas contraire, sauf si les modèles sont définis à partir de critères exclusifs mettant en jeu des postes capacitaires (ou non capacitaires) différents, les formulations ne sont pas équivalentes.

- **Contraintes $1/(\nu+\mu)$**. Dans la pratique, ces relations 4 se retrouve le plus souvent avec $\nu_k = 1$; les contraintes d'ordonnancement sont alors du type « $1/(\nu+\mu)$ »: «pas plus d'un toit ouvrant sur 5 véhicules consécutifs» ($\mu_k = 4$).
- **Simplification possible du problème général**. Si la variété des temps opératoires sur le poste *k* est supérieure à deux, il est possible de se ramener à ce cas en retenant comme temps opératoire de chaque critère *f* de l'ensemble $\mathcal{F}_k$ des critères affectant la durée de travail de ce poste *k* : $T_k^{max} = \underset{f}{Max}\{\theta_{fk}\}$ et, $T_k^{min} = \underset{\{f \in \mathcal{F}_k \mid \theta_{fk} < \bar{\theta}\}}{Max}\{\theta_{fk} < \bar{\theta}\}$, dans le cas contraire. Cette simplification permet de limiter la taille du problème (usage des relations 4 au lieu des relations 3) mais elle contraint le problème d'ordonnancement plus que nécessaire.
- **Ségrégation de flux sur deux brins ligne.** Ce type de contrainte peut être utilisé pour gérer la ségrégation des flux lorsqu'une ligne se partage momentanément en deux pour permettre de traiter une quantité de travail plus importante sur un brin que sur l'autre, la ségrégation s'appuyant sur un critère mobilisant des processus techniques différents. Par exemple, en imposant que sur 3 véhicules consécutifs on ait un seul véhicule au plus ayant le critère A, on permet de dériver les véhicules ayant cette caractéristique A vers le brin de ligne α et les autres véhicules vers le brin de ligne $\bar{\alpha}$, les deux flux fusionnant ensuite au niveau du poste sur lequel les deux brins convergent, en respectant l'ordre initial. Cette synchronisation implique que la cadence d'entrée sur chacun des postes de travail de l'un des brins soit celle de la ligne (par exemple le brin $\bar{\alpha}$ comportant $\kappa_{\bar{\alpha}}$ postes). Sur l'autre brin (α), le nombre de postes peut être plus faible ($\kappa_\alpha < \kappa_{\bar{\alpha}}$), pour une charge de travail identique ($\bar{\theta} \cdot \kappa_{\bar{\alpha}}$), ce qui permet d'avoir un temps de cycle plus élevé ($\theta_\alpha \leq \theta \cdot (\kappa_{\bar{\alpha}}/\kappa_\alpha)$) en conservant la synchronisation. des flux[1]. Le non-respect de cette contrainte provoque nécessairement un arrêt de la ligne par saturation, le véhicule ayant le critère A et violant la contrainte étant obligé d'attendre la libération du premier poste du brin de ligne α si celui-ci doit exécuter un travail d'une durée égale
- **Limitation des séquences de véhicules ayant le même critère**. Ces formulations de contrainte d'espacement peuvent être utilisées pour éviter que l'ordonnancement ne conduise à des séries trop longues de véhicules présentant un même critère ; dans ce cas, ce n'est pas la variabilité des temps opératoires qui justifie la contrainte d'espacement mais des raisons techniques (sur-chauffe…). Par exemple, en imposant que sur 10 véhicules consécutifs on ait un seul véhicule au plus ayant le critère A (critère fictif destiné à limiter la séquence), on limite à 9 la plus longue séquence de véhicules n'ayant pas ce critère.

[1] Exemple: $\theta_{\bar{\alpha}} = \theta = 1$, $\kappa_{\bar{\alpha}} = 8$, $\kappa_\alpha = 2 \Rightarrow \theta_\alpha \leq 4$. Le temps de cycle du brin α peut donc être 300% supérieur à celui du brin $\bar{\alpha}$ (et donc du reste de la ligne). En réalité, c'est le temps de cycle θ_α du brin α qui conditionne l'espacement, le nombre minimal de véhicules de critère $\underline{A}$ devant s'intercaler dans l'ordonnancement entre deux véhicules de critère A étant l'arrondi supérieur du rapport $(\theta_\alpha - \theta_{\bar{\alpha}})/\theta_{\bar{\alpha}}$ (= $(\theta_\alpha - \theta)/\theta$), sachant que ce temps de cycle θ_α et le nombre de postes κ_α et $\kappa_{\bar{\alpha}}$ concernés par cette scission du flux doivent conduire à une même charge de travail sur chacun des brins et à effectuer des opérations similaires sur les postes suivants.

- **Contraintes sur critères croisés.** Cette formulation de contrainte d'espacement peut être adaptée pour forcer un véhicule possédant le critère B à être suivi d'au moins μ_{BA} véhicules ne possédant pas ce critère B avant de trouver un véhicule possédant le critère A, ces deux critères étant exclusifs[1]. On parle alors de contraintes sur **critères croisés**, ces contraintes étant utilisées depuis des années dans l'industrie automobile. Elles sont utilisées notamment lorsque le changement de critère amène l'opérateur du poste à changer d'outillage normalement en temps masqué, ce qui n'est possible que si un nombre suffisant de véhicules sépare le véhicule possédant le critère A, du véhicule possédant le critère B. En général on a $\mu_{BA} = 1$. Si la contrainte est symétrique[2], il faut poser une seconde contrainte. Examinons comment adapter les contraintes des relations 4 en nous intéressant aux véhicules de rang $h > \mu_{BA}$. Le véhicule de rang $j < h$ ne possède pas le critère B si $\sum_{i=1}^{N} \delta_{iB} \cdot x_{ij} = 0$. Le problème est alors que, sauf spécification contraire, on a le droit de faire se succéder des véhicules ayant le critère B si le véhicule de rang *h* et, le cas échéant, plusieurs véhicules suivants ne possèdent pas le critère A. Pour contourner cette difficulté, il faut introduire la variable indicatrice y_{hA} valant 1 si le véhicule de rang *h* sur le poste *k* considéré possède le critère A et 0 dans le cas contraire $y_{hA} = \sum_{i=1}^{N} \delta_{iA} \cdot x_{ij}$. L'interdiction recherchée est alors obtenue par les relations 5.

$$\sum_{j=[h-r]}^{h-1} \sum_{i=1}^{N} \delta_{iB} \cdot x_{ij} < (1 - y_h)\mu_{BA},\ y_{hA} = \sum_{i=1}^{N} \delta_{iA} \cdot x_{ij}$$

$$\textit{pour } h = \mu_{BA} + 1, \ldots, \mathrm{N};\ \mathrm{r} = 1, \ldots, \mu_{BA} \textit{ et } k \in \mathcal{K}_1 \qquad \textit{relations 5}$$

En effet, si le véhicule de rang *h* possède le critère A, alors $(1 - y_h)\mu_{BA} = 0$ et les μ_{BA} premières contraintes interdisent la présence de véhicules possédant le critère B et, dans le cas contraire, $(1 - y_h)\mu_{BA} = \mu_{BA}$ et les μ_{BA} premières contraintes sont toujours satisfaites que les μ_{BA} véhicules précédents possèdent ou non le critère B. Si des contraintes d'espacement portent sur le critère B, elles sont exprimées par l'adaptation des relations 4 en remplaçant A par B dans ces relations, et la conjonction de ces contraintes assure le respect des contraintes d'espacement sur B et les contraintes croisées d'espacement de B par rapport à A (non symétriques rappelons-le).

[1] Exemple : «un véhicule présentant le critère B ne peut suivre un véhicule possédant le critère A qu'à condition que 3 véhicules ne possédant pas le critère B les séparent»). Les séquences $B - \overline{B} - \overline{B} - \overline{B} - A$ et $B - \overline{B} - \overline{B} - \overline{B} - \overline{B} - A$ sont autorisées mais la séquence $B - \overline{B} - \overline{B} - A$ est interdite

[2] Pour reprendre le même exemple : «un véhicule présentant le critère A ne peut suivre un véhicule possédant le critère B qu'à condition que 3 véhicules ne possédant pas le critère A les séparent».

- **Prise en compte de l'ordonnancement glissant**. Le processus de production sur une ligne est continu : le soir on arrête le travail sur la ligne, pour reprendre le lendemain matin le travail là où il s'était arrêté. Pour apprécier l'impact de l'ordonnancement glissant, il faut expliciter de manière plus précise ce qui se passe dans le temps et dans l'espace, d'autant plus que cette mise au point sera indispensable pour l'analyse des renforts (§ 3.2.2.2, page 251). Jusqu'ici, seul le rang d'arrivée d'un véhicule sur le poste *k* a été pris en considération, le passage du rang au numéro de période de traitement du véhicule s'effectuant immédiatement en raison de la constance du temps de cycle. Ce repérage temporel est «local», le traitement de ce véhicule pouvant se faire le jour du lancement ou le lendemain. En effet, l'ordonnancement décidé pour une journée (période arbitraire utilisée ici pour fixer les idées) ne produit ses effets qu'en partie au cours de la journée. Supposons que la ligne se caractérise par un temps de cycle d'une minute ($\bar{\theta} = 1$), qu'elle comporte K = 200 postes (⇒ charge de travail de 200 minutes) et que le temps quotidien d'ouverture de la ligne soit de T = 700 minutes, ce qui conduit à un ordonnancement quotidien de 700 véhicules sur la ligne. En début de journée, au lancement du premier véhicule (première décision mise en œuvre pour le nouvel ordonnancement), les K = 200 postes de travail de la ligne sont occupés, le premier par le premier véhicule ordonnancé et les K – 1 = 199 postes suivants par les K – 1 derniers véhicules ordonnancés la veille ; les K – 1 derniers véhicules de l'ordonnancement du jour seront donc traités en partie ce jour et en partie le lendemain[1]. L'ordonnancement des N véhicules pour la nouvelle journée doit nécessairement tenir compte des décisions prises la veille pour respecter correctement les contraintes d'espacement pour chaque critère et les contraintes croisés. Il s'ensuit que pour les contraintes définies par les relations 4 et 5 sont à adapter pour les premiers véhicules à placer. Par exemple, les relations 4 étaient définies pour $h = \nu_k + \mu_k, \ldots, \mathrm{N}$. En pratique, elles sont définies pour $h = 1, \ldots, \mathrm{N}$ mais les $\nu_k + \mu_k$ premières contraintes sont liées aux décisions de la veille, lesquelles conduisent à avoir déjà en amont du poste *k* une séquence de véhicules lancés la veille[2].

[1] Le dernier véhicule des 700 véhicules ordonnancés le jour J passe sur le premier poste de la ligne en fin de cette journée J et passera sur les 199 postes restants au cours de la journée J + 1. Le véhicule ayant le rang 501 est le dernier des véhicules ordonnancé le jour J et à être traité en totalité à la fin du jour J.

[2] Supposons par exemple que $\nu_k = 1$, $\mu_k = 3$ et que sur les trois derniers véhicules ordonnancés, seul celui du milieu possède le critère A. Dans ces conditions, on a :

- pour $h = 1$: $0 + 1 + 0 + \sum_{i=1}^{\mathrm{N}} \delta_{i\mathrm{A}} \cdot x_{i1} < 1 \Rightarrow \delta_{i_f\mathrm{A}} \cdot x_{i_f1} = 0$ ⇒ interdiction de commencer l'ordonnancement par un véhicule possédant le critère A.
- pour $h = 2$: $1 + 0 + \sum_{j=1}^{2} \sum_{i=1}^{\mathrm{N}} \delta_{i\mathrm{A}} \cdot x_{ij} < 1 \Rightarrow \delta_{i_f\mathrm{A}} \cdot x_{i_f2} = 0$ ⇒ le deuxième véhicule ne peut également pas posséder le critère A.
- pour $h = 3$: $0 + \sum_{j=1}^{3} \sum_{i=1}^{\mathrm{N}} \delta_{i\mathrm{A}} \cdot x_{ij} < 1$ ⇒ le troisième véhicule peut ou non posséder le critère A.
- pour $h > 3$ on repart sur la relation générale : $\left(\sum_{j=h-3}^{h} \sum_{i=1}^{\mathrm{N}} \delta_{i\mathrm{A}} \cdot x_{ij} < 1 \right)$.

3-2.2 Prise en compte des contraintes des postes non capacitaires

Comme on l'a indiqué, les postes non capacitaires ($k \in \mathcal{K}_2$) sont soumis aux mêmes contraintes que les postes capacitaires à la différence près que ces contraintes peuvent être violées parce qu'il est possible d'accroître momentanément la capacité du poste de travail pour faire face à la surcharge de travail. Le personnel utilisé en renfort est amené à se déplacer en fonction de la localisation dans le temps et dans l'espace de ces pointes d'activité et des limites de polyvalence qui le caractérise. Il prend alors en charge un nouveau véhicule pendant que le titulaire du poste achève le travail en cours sur le véhicule précédent.

Ces violations ont donc un coût imputable à la fois aux caractéristiques de l'ensemble à ordonnancer mais aussi de la qualité de l'ordonnancement. Il est alors nécessaire d'introduire une fonction économique d'évaluation permettant de qualifier les ordonnancements possibles. Le problème passe alors de la définition d'un ordonnancement admissible, à celui d'un ordonnancement admissible minimisant une fonction de coût. L'introduction d'une fonction-objectif dans la formalisation du problème se fait à ce stade de la formalisation générale du problème. Elle sera complétée ensuite avec la prise en compte de l'incidence de l'ordonnancement sur d'autres coûts (§ 3.3, page 254). On commencera par décrire la transformation des contraintes (§ 3.2.2.1) avant de déterminer le nombre de renforts découlant de l'ordonnancement (§ 3.2.2.2, page 251) et d'en tirer les conséquences sur la fonction économique (§ 3.2.2.3, page 253).

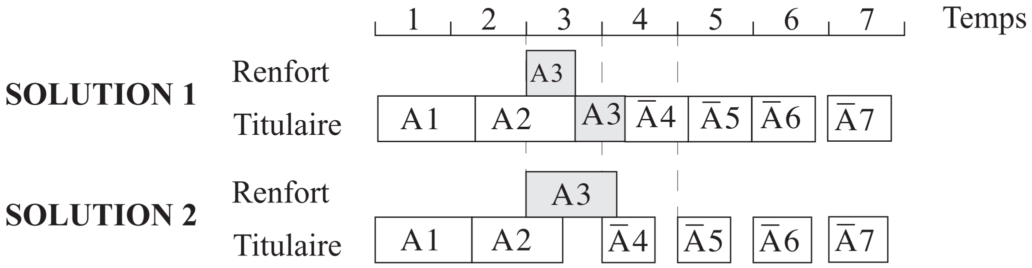
3.2.2.1 Adaptation des contraintes d'espacement

Le raisonnement qui a conduit aux relations 4 implique que si les ν_k premiers véhicules possèdent le critère A et que le véhicule de rang $\nu_k + 1$ possède également ce critère, le retard maximal R_k^{max} que ce poste peut absorber est dépassé, l'excédent de charge étant $\mathrm{T}_k^{max} - \bar{\theta}$. A priori, ce dépassement peut être pris en charge par un opérateur en renfort, le titulaire du poste étant occupé par ailleurs, jusqu'au moment où l'excédent est résorbé, ce qui se produit au bout de $(\mathrm{T}_k^{max} - \bar{\theta})/(\bar{\theta} - \mathrm{T}_k^{min})$ cycles, après quoi le titulaire du poste peut éventuellement prendre la relève du renfort pour achever le travail que celui-ci avait commencé. Pour bien comprendre ce qui se passe physiquement et expliquer les alternatives organisationnelles que l'on a, un exemple est nécessaire.

- Supposons que $\mathrm{T}_k^{max} = 80$, $\mathrm{T}_k^{min} = 50$, $\bar{\theta} = 60$ et $R_k^{max} = 40$. Il s'ensuit que $\nu_k = 2$ et $\mu_k = 2$. La surcharge de travail (20) occasionnée par un véhicule ayant le critère A est absorbée par la sous-charge de travail (10) des deux premiers véhicules suivants ne possédant pas le critère A.
- En supposant que les 4 derniers véhicules ordonnancés la veille soient tous sans le critère A, la séquence $A - A - \bar{A} - \bar{A} - \bar{A} - \bar{A}$ respecte la contrainte d'espacement et l'évolution de la charge de travail excédentaire à la fin des 6 premiers cycles est $\{20; 40; 30; 20; 10; 0\}$. À partir du moment où la charge de travail excédentaire retombe à 20, le véhicule suivant peut à nouveau posséder le critère A, la contrainte «pas plus de deux véhicules possédant le critère A dans une séquence de 4 véhicules» étant respectée, à condition que les deux véhicules suivants ne possèdent

pas ce critère, autrement dit, la séquence $A - A - \bar{A} - \bar{A} - A - \bar{A}$ est admissible (à condition que le 7e véhicule ne possède pas le critère A).

- Avec la séquence $A - A - A - \bar{A} - \bar{A} - \bar{A}$, l'évolution de la charge de travail excédentaire à la fin des 6 premiers cycles est $\{20; 40; 60; 50; 40; 30\}$, ce qui nécessite la présence d'un renfort au cours des périodes 3 et 4 avant de retrouver le report maximal admissible. Le Gantt des ressources permet de visualiser deux solutions physiquement possibles.



La première solution conduit à un partage du travail à effectuer sur le troisième véhicule et à l'évolution suivante de la charge de travail excédentaire : $\{20; 40; 20$[1]$;$ $10; 0; 0\}$. À la fin de la troisième période, tout se passe comme si, pour le choix des véhicules suivants, on recommençait l'ordonnancement sans que les véhicules précédents (1 et 2) possèdent le critère A (le troisième véhicule possédant ce critère). C'est vrai ici parce que le report maximal n'excède pas le temps opératoire maximal $(T_k^{max} \leq R_k^{max})$. Si cette condition est remplie, toute surcharge se traduit par une réinitialisation des contraintes. Dans le cas contraire, la contrainte d'espacement reste la même.

La seconde solution est sans doute plus réaliste car le partage du travail entre deux opérateurs n'est pas sans problème, en particulier lorsque le temps de cycle est court. De surcroît, elle desserre les contraintes d'ordonnancement comme nous allons le voir. Avec cette solution, l'évolution de la charge de travail excédentaire est : $\{20; 40; 0; 0; 0; 0\}$. À la fin de la troisième période, tout se passe donc comme si, pour le choix des véhicules suivants, on recommençait l'ordonnancement sans qu'aucun des véhicules ordonnancés avant (1, 2 et 3) ne possèdent le critère A, ce qui est vrai ici parce que le report maximal n'excède pas le temps de cycle $(T_k^{max} \leq \bar{\theta})$. Dans le cas contraire, tout se passe comme si, pour le choix du quatrième véhicule et les suivants, on recommençait l'ordonnancement sans que les véhicules précédents (1 et 2) possèdent le critère A (le troisième véhicule possédant ce critère).

À partir de ces observations, il est possible d'adapter les contraintes des relations 4. L'hypothèse organisationnelle que l'on retiendra est celle correspondant à la seconde solution parce qu'elle semble la plus réaliste (sinon, il faut adapter les relations 3). On

[1] $80 - 60 = 20$

introduit alors la variable z_{hk} qui vaut 1 si, sur le poste k non capacitaire, on ordonnance en rang h un véhicule possédant le critère A et violant la contrainte du report cumulé maximal. L'impact de cette variable sur la fonction-objectif la conduit à prendre la valeur 0, sauf en cas de violation de contrainte.

On retiendra pour commencer le cas $(\mathrm{T}_k^{max} \le \bar{\theta})$, qui conduit à une remise à zéro de l'excédent de charge de travail R_{kh} à résorber après traitement du véhicule de rang h lorsque celui-ci implique la présence d'un renfort. Ceci se traduit, dans la formulation des relations 4, par une remise à zéro du nombre de véhicules ayant le critère A $\sum_{j=[h-(\nu_k+\mu_k)+1]}^{h}\sum_{i=1}^{\mathrm{N}}\delta_{ik}\cdot x_{ij}$, sur la fenêtre glissante de $(\nu_k+\mu_k)$ véhicules de rang inférieur ou égal à h. Ce nombre pouvant être remis à zéro, il faut introduire la variable v_{hk} correspondant à ce cumul glissant, ce qui conduit aux relations 6.

- $v_{hk} = \sum_{j=1}^{(\nu_k+\mu_k)}\sum_{i=1}^{\mathrm{N}}\delta_{ik}\cdot x_{ij}$*, pour* $h = \nu_k+\mu_k$ *et* $k \in \mathscr{K}_2$ *relations 6*

- $v_{hk}+z_{hk} = v_{(h-1)k}+\sum_{i=1}^{\mathrm{N}}\delta_{ik}\cdot x_{ih}-\sum_{i=1}^{\mathrm{N}}\delta_{ik}\cdot x_{i(h-\nu_k+\mu_k+1)}$,

  pour $h = (\nu_k+\mu_k)+1$ *et* $k \in \mathscr{K}_2$ *(relations 6 - suite)*

  Dans le premier membre de cette relation, la variable z_{hk} correspond à un dépassement de capacité par mobilisation de renfort si elle est amenée à prendre la valeur 1.

- $v_{hk}+z_{hk} = v_{(h-1)k}\cdot(1-z_{(h-1)k})$

$$+\sum_{i=1}^{\mathrm{N}}\delta_{ik}\cdot x_{ih}-\left(\sum_{i=1}^{\mathrm{N}}\delta_{i\mathrm{A}}\cdot x_{i(h-\nu_k+\mu_k+1)}-s_{hk}\right)$$

$$\textit{avec } s_{hk} = \left(\sum_{i=1}^{\mathrm{N}}\delta_{ik}\cdot x_{i(h-\nu_k+\mu_k+1)}\right)\sum_{j=h-(\nu_k+\mu_k)+1}^{h-1}v_{jk}$$

et $s_{hk}\le 1$ *pour* $h > (\nu_k+\mu_k)+1$ *et* $k \in \mathscr{K}_2$ *(relations 6 - fin)*

Le premier membre de cette dernière relation est identique à celui de la relation précédente; justifions les termes du second membre. Le premier correspond au report de la période précédente, s'il est positif. Le second correspond au booléen du critère du véhicule venant d'être choisi au rang h (ce qui ajoute 1 au cumul glissant si ce véhicule possède ce critère). On retranche ensuite le booléen du critère du dernier véhicule de la fenêtre glissante précédente, mais ce dernier terme n'est pas correct si au moins un renfort a été mobilisé pour un véhicule de rang supérieur à celui du dernier véhicule de la fenêtre glissante précédente et donc pour les rangs

$h-(\nu_k+\mu_k)+1$ à $h-1$. La variable s_{hk} permet de ne retrancher qu'à bon escient le booléen du critère du dernier véhicule de la fenêtre glissante précédente. Dans la définition de cette variable s_{hk}, l'expression $\sum_{j=h-(\nu_k+\mu_k)+1}^{h-1} v_{jk}$ correspond au nombre de renforts mobilisés au cours des $(\nu_k+\mu_k-1)$ périodes précédentes; s_{hk} ne peut prendre cette valeur, parce que ce correctif n'intervient pas si le booléen du critère du dernier véhicule de la fenêtre glissante précédente est nul et parce que, dans le cas contraire il ne peut excéder 1; la première observation est prise en compte par la multiplication de ce nombre par ce booléen. La seconde observation est prise en compte par la contrainte $s_{hk} \leq 1$ qui implique que l'on interdise d'avoir deux renforts travaillant simultanément en plus du titulaire du poste, ce qui semble réaliste et assure un certain lissage dans l'utilisation des renforts (lever cette contrainte implique seulement la création d'une nouvelle variable).

Avant d'examiner l'impact économique de l'usage des renforts il convient d'abord de pouvoir les décompter et donc de pouvoir déterminer l'évolution de la charge temporelle des véhicules, ce qui oblige à passer d'une logique locale d'analyse d'un poste, à une logique globale d'analyse simultanée de l'ensemble des postes et implique d'utiliser un même repérage temporel.

3.2.2.2 Détermination du nombre de renforts

L'analyse du problème de l'ordonnancement glissant (page 247) a permis d'introduire la relation entre le rang h (h = 1,… , N) d'un véhicule, le poste k (k = 1,… , K) sur lequel il se trouve et la date absolue t (t = 1,… , T). La demande de renfort pour une journée donnée dépend donc de la position des postes non capacitaires. Le véhicule de rang h dans l'ordonnancement du jour J arrive sur le poste k au début de la période $t = k+h-1$ au cours de la journée J si $k+h-1 \leq \mathrm{T}$ et le jour J + 1, dans le cas contraire. Par exemple, pour K = 200, T = N = 700, une décision d'ordonnancement du véhicule h =600, sur le poste k = 150 (poste supposé non capacitaire) sera exécutée au cours de la période t = 150 + 600 – 1 = 749, c'est-à-dire au cours de la 49e minute du jour J + 1. Si cette décision implique un renfort, celui-ci n'interviendra que le lendemain de la prise de décision. Certains renforts mobilisés au cours d'une journée auront donc été décidés la veille, tandis que d'autres décidés aujourd'hui ne seront mis en place que le lendemain.

L'opérateur travaillant en renfort est nécessairement amené à partager son temps entre plusieurs postes non capacitaires (dans la pratique, rarement plus d'une demi-douzaine). Ce qui complique l'analyse économique de la décision, c'est que les renforts mobilisés pour une journée sont requis à la suite de décisions prises la veille et d'autres, de décisions prises le jour même[1]. Cet aspect du problème est incontournable car il découle de l'ordonnancement glissant, consistant à générer périodiquement un ordonnancement d'un nouvel ensemble de véhicules, sans attendre que soit complètement traité l'ensemble des véhi-

[1] Pour reprendre notre exemple, en J + 1, on a un renfort sur le poste k = 150, à partir de la période 49, Ce même renfort peut être appelé pour travailler sur le poste non capacitaire k' = 100, à partir de la période 115, ce qui résulte de l'ordonnancement décidé pour J + 1.

cules précédemment ordonnancé. Dans l'analyse économique des renforts, trois positions, au moins, peuvent être adoptées:

- On peut vouloir évaluer l'incidence économique exacte des renforts effectivement mobilisés au cours d'une journée. Cette option conduit à considérer comme des données les conséquences des décisions de renfort prises la veille et exécutées au cours de la journée. Dans cette optique, les décisions de renfort qui ne seront exécutées que le lendemain, ne sont pas prises en compte dans l'évaluation économique.
- L'inconvénient de l'option précédente est de ne pas soumettre à une évaluation économique toutes les décisions prises le jour J et d'inciter à un report de décisions de renfort vers la fin de l'ordonnancement, là où l'appel à ces renforts n'est pas pénalisé. Ce choix fait donc courir le risque de dégrader l'évaluation de l'ordonnancement du lendemain au profit de celui du jour. C'est pourquoi on peut préférer une formulation hybride dans laquelle l'évaluation économique ajoute au coût des renforts effectivement supportés pour le jour pour lequel l'ordonnancement est établi, la partie du coût des renforts du lendemain résultant de cet ordonnancement. Cet «effet de bord» est d'autant moins marqué que le nombre N de véhicules à ordonnancer est fort par rapport au nombre total K de postes.
- Une troisième option est envisageable. Elle consiste à considérer que ce problème est trop compliqué pour être pris en compte dans une formulation globale, d'autant qu'il faut tenir compte dans le calcul des renforts, de leurs déplacements d'un poste à un autre. Dans ces conditions, la minimisation du nombre d'appels de renforts va dans le sens d'une meilleure efficacité économique.

Si aucune de ces positions n'est entièrement satisfaisante, le choix de l'une d'entre elles est nécessaire pour introduire un point de vue économique permettant d'évaluer des alternatives décisionnelles à travers la fonction-objectif. La seconde de ces positions est préférable à l'évidence à la première. Son choix par rapport à la troisième se justifie par le fait que la fonction-objectif devra ultérieurement inclure d'autres éléments de coût, des arbitrages économiques pouvant être à faire entre des purges dans l'atelier de peinture et des renforts. La seconde solution évite l'arbitraire dans la définition des coefficients de la fonction-économique et donne un ordre de grandeur des enjeux économiques liés à l'ordonnancement.

On a vu au § 3.2.1.2, page 243, que le véhicule de rang *h* dans l'ordonnancement du jour J arrive sur le poste *k* au début de la période $t = k + h - 1$ au cours de la journée J si $k + h - 1 \leq \mathrm{T}$ et le jour J + 1, dans le cas contraire. Au début de la période *t*, on traite donc sur le poste *k* le véhicule de rang $t - k + 1$. Compte tenu des hypothèses organisationnelles retenues (solution 2 du Gantt), ce renfort est présent sur un nombre de cycles correspondant à la partie entière inférieure de $\{\mathrm{T}_k^{max}/\bar{\theta}\}$, que l'on notera β_k. Ce renfort introduit en *t* sur le poste *k* non capacitaire $(k \in \mathscr{K}_2)$ est donc présent sur ce poste des périodes *t* à $t + \beta_k - 1$, ce qui revient à dire qu'un renfort présent au cours de la période *t* a commencé son travail au début de *t*' compris entre $t - \beta_k + 1$ et *t*.

Si l'on ne se préoccupe pas des effets de bord de l'ordonnancement glissant, le cumul du nombre de renforts présents au cours de la période *t* est donc

$\sum_{k \in \mathcal{K}_2} \sum_{i=1}^{N} \delta_{ik} \sum_{t' = t - \beta_k + 1}^{t} x_{i,\, t-k'+1}$, sachant que les hypothèses organisationnelles retenues limitent à 1 ce nombre de renforts à une période quelconque.

On a vu que les véhicules de rang h dans l'ordonnancement arrivent sur le poste k le jour J + 1, si $t = k + h - 1 > \mathrm{T}$. Les éventuels renforts du jour J + 1 sont donc pris en compte dans l'expression $\sum_{k \in \mathcal{K}_2} \sum_{i=1}^{N} \delta_{ik} \sum_{t' = t - \beta_k + 1}^{t} x_{i,\, t-k'+1}$. Par contre, cette expression ne tient pas compte de ceux décidés la veille sur le poste k et qui obéissent à la même condition $k + h - 1 > \mathrm{T}$. On notera γ_{kt} ces renforts décidés la veille.

Si l'on suppose que la journée est partagée en deux équipes travaillant chacune pendant T/2 minutes et s'il est vraisemblable que tout travail de renfort commencé la veille est achevé la veille, le nombre de renforts z_1, z_2 de chacune de deux demi-journées du jour J, dont les valeurs auront tendance à être les plus basses possibles à cause de la fonction-objectif, est donc tel que:

$$z_1 \geq \sum_{k \in \mathcal{K}_2} \sum_{i=1}^{N} \delta_{ik} \sum_{t' = t - \beta_k + 1 > 0}^{t} x_{i,\, t-k'+1} + \gamma_{kt},\ \textit{pour } t = 1, \ldots, \mathrm{T}/2$$

$$z_2 \geq \sum_{k \in \mathcal{K}_2} \sum_{i=1}^{N} \delta_{ik} \sum_{t' = t - \beta_k + 1}^{t} x_{i,\, t-k'+1} + \gamma_{kt},\ \textit{pour } t = \mathrm{T}/2 + 1, \ldots, \mathrm{T} \qquad \textit{relations 7}$$

Le jour J + 1, des renforts découleront de l'ordonnancement décidé le jour J. On a vu que ceux-ci découlaient des décisions prises pour les véhicules de rang h sur le poste k pour $k + h - 1 > \mathrm{T}$. On peut alors reprendre la relation précédente pour des indices de période $t > \mathrm{T}$ **:**

$$z_3 \geq \sum_{k \in \mathcal{K}_2} \sum_{h=1}^{N} \sum_{i=1}^{N} \delta_{ik} \sum_{t' = t - \beta_k + 1 > 0}^{t} x_{i,\, t-k'+1},\ \textit{pour } t = \mathrm{T} + 1, \ldots, \mathrm{T} + \mathrm{K},$$
$$\textit{avec } k + h - 1 \leq \mathrm{T} \qquad \textit{relation 8}$$

On notera que, dans la fonction-objectif que nous allons introduire, les temps de transport d'un poste non capacitaire à un autre ne sont pas pris en compte. Celle-ci est formellement possible avec une matrice de temps de transport d'un poste non capacitaire à un autre mais elle complique la formulation en obligeant à assigner les opérateurs en renfort aux différents renforts (formulation dérivée de formulations classiques).

3.2.2.3 Introduction de la fonction économique

La fonction-objectif correspond au coût des renforts. Pour un coût unitaire de renfort γ, la fonction-objectif est

$$Min[\gamma_1(z_1 + z_2 + z_3)] \qquad \textit{relation 9}$$

3.3 Prise en compte des lots de peinture

Les véhicules passent tous par un atelier de peinture. Le changement de couleur dans l'atelier implique une purge des tuyaux et des pistolets, ce qui a un coût γ_2. Le coût d'une purge peut être ou non dépendant de la couleur précédente. Il semblerait que dans l'industrie automobile, l'hypothèse d'indépendance soit suffisante. Par ailleurs, des contraintes techniques d'encrassement conduisent la taille d'une séquence de véhicules ayant la même couleur (on parle encore de rafale de teinte) à ne pas dépasser un seuil P. On se restreindra ici à un seul poste de peinture mais la généralisation est immédiate et peut concerner d'autres postes pour lesquels un changement fréquent d'outillages ou de réglages génère des coûts que l'on désire limiter.

Notons u_h la variable dichotomique valant 1 si le véhicule de rang *h* a une teinte différente de celle du véhicule précédent ($h - 1$). Dans ces conditions, la fonction- objectif de la relation 10 doit inclure le coût des purges et devient:

$$Min\left[\gamma_1(z_1 + z_2 + z_3) + \gamma_2 \sum_{h=2}^{N} u_h\right] \qquad \textit{relation 10}$$

Il faut forcer la variable u_h à prendre la valeur 1 lorsqu'il y a changement de réglages sur le poste *k*. Notons π_i le numéro de teinte utilisé par le véhicule *i* (avec $\pi_i < M$). Le numéro de teinte prise par le véhicule de rang *h* dans l'ordonnancement est $\sum_{i=1}^{N} \pi_i \cdot x_{ih}$.
Le changement de réglage entre le véhicule $h - 1$ et le véhicule *h* arrive lorsque la différence $\sum_{i=1}^{N} \pi_i \cdot x_{ih} - \sum_{i=1}^{N} \pi_i \cdot x_{i(h-1)}$ n'est pas nulle. Pour forcer u_h à prendre la valeur 1 en cas de changement, il faut introduire la contrainte suivante:

$$-Mu_h \leq \sum_{i=1}^{N} \pi_i \cdot x_{ih} - \sum_{i=1}^{N} \pi_i \cdot x_{i(h-1)} \leq Mu_h, \text{ pour } h = 2, \ldots, N \qquad \textit{relation 11}$$

qui conduit à l'effet recherché parce que la fonction-objectif tend à rendre nuls le plus possible de u_h et que les u_h de la relation 11 ne peuvent être nuls que si numéro de réglage du poste *s* ne change pas et que dans les autres cas (second terme strictement positif ou négatif), u_h est nécessairement égal à 1.

Il convient enfin d'ajouter la contrainte glissante relative au nombre maximal de purges qui est donnée par la relation 12.

$$\sum_{h=j}^{j+P-1} u_h \geq 1 \; pour\, j = 1, \ldots, N - P + 1 \qquad \textit{relation 12}$$

4 Conclusion

L'abondance de la littérature relative au problème d'ordonnancement de produits hétérogènes sur ligne d'assemblage témoigne de l'intérêt que suscite un tel problème dans un contexte de production d'une diversité sans cesse renouvelée. Un examen plus minutieux de cette littérature en révèle néanmoins les faiblesses qui prennent leur source dans les hypothèses qui sous-tendent ces modélisations et en limitent substantiellement le champ d'application. Le lancement en production implique l'usage de méthodes de séquencement prenant en compte la totalité des points de vue, qu'ils émanent de la tôlerie, de la peinture ou de l'assemblage. La modélisation proposée dans cet article représente à cet égard une avancée pour la compréhension et la résolution de cette classe de problèmes. Elle nous semble constituer un point de départ important pour la construction de méthodes heuristiques, la résolution exacte semblant compromise pour le traitement de problèmes de taille réelle en raison de l'importance des temps de calcul qu'une telle résolution exigerait. L'élaboration d'une approche heuristique est actuellement en cours et produit des résultats prometteurs sur des instances de grande taille.

5 Bibliographie

[1] Bautista, J., Companys, R., Corominas, A., «Note on cyclic sequences in the product rate variation problem», *European Journal of Operational Research*, 2000, 468-477.

[2] Bautista, J., Companys, R., Corominas, A., «Heuristics and exact algorithms for solving the Monden problem», *European Journal of Operational Research*, 1993, 101-113.

[3] Choi, W., Shin, H., «A real-time sequence control system for the level production of the automobile assembly line», *Computers and Industrial Engineering*, 1997, 769-772.

[4] Ding, F.-Y., Cheng, L., «A effective mixed-model assembly line sequencing for Just-In-Time production systems», *Journal of Operations Management*, 1993, 45-50.

[5] Giard, V., *Processus productifs et programmation linéaire*, Economica, 1997.

[6] Giard, V., *Gestion de la production et des flux*, 3e éd., Economica, 2003.

[7] Gottlieb, J., Puchta, M., Solnon, C., «A study of greedy, local search and ant colony optimization approaches for car sequencing problems, in Applications of Evolutionary Computing», *Lecture Notes in Computer Science*, Springer, 2003, 246-257.

[8] Hyun, C. J., Kim, Y., Kim, Y. K., «A genetic algorithm for multiple objective sequencing problems in mixed model assembly line», *Computers and Operations Research*, 1998, 675-690.

[9] Kim, Y. K., Hyun, C. J., Kim, Y, «Sequencing in mixed model assembly lines : a genetic algorithm approach», *Computers and Operations Research*, 1996, 1131-1145.

[10] Korkmazel, T., Merel, S., «Bicriteria sequencing methods for the mixed-model assembly line in just-in-time production systems», *European Journal of Operational Research*, 2001, 188-207.

[11] Kubiak, W., Sethi, S., «A note on level schedules for mixed-model assembly lines in just-in-time production systems», *Management Science*, 1991, 121-122.

[12] Kubiak, W., Sethi, S., «Optimal just-in-time schedules for flexible transfer lines», *Journal of Flexible Manufacturing Systems*, 1994, 137-154.

[13] Leu, Y., Matheson, L. A., Rees, L. P., «Sequencing mixed model assembly lines with genetic algorithms», *Computers and Industrial Engineering*, 1996, 1027-1036.

[14] Miltenburg, J., Steiner, G., Yeomans, S., «Dynamic programing algorithm for scheduling mixed-model just-in-time production systems», *Mathematical Computing and Modelling*, 1990, 57-66.

[15] Mitsumori, S., «Optimal schedule control of conveyor line», *IEEE Trans. Auto. Control*, 1969, 633-639

[16] Monden, Y., *Toyota Production System : An Integrated Approach to Just-In-Time*, 3e éd., Engineering & Management Press, 1998.

[17] Murata, T., Ishibuchi, H., Tanaka, H, «Multi-objective algorithm and its application to flow shop scheduling», *Computers and Industrial Engineering*, 1996, 957-968.

[18] Smith, K., Palaniswami, M., Krishnamoorthy, N., «Traditional heuristic versus Hopfield neural network approaches to a car sequencing problem», *European Journal of Operational Research*, 1996, 300-316.

[19] Sumichrast, R. T., Russell, R. S., Taylor, B. W., «A comparative analysis of sequencing procedures for mixed-model assembly lines in a just-in-time production system», *International Journal of Production Research*, 1992, 199-214.

[20] Tamura, T ., Long, H., Ohno, K., «A sequencing problem to level part usage rates and work loads for a mixed-model assembly line with a bypass subline», *International Journal of Production Economics*, 1999, 557-564.

[21] Xiaobo, Z., Ohno, K., «Algorithms for sequencing mixed models on an assembly line in a JIT production system», *Computers and Industrial Engineering*, 1997, 47-56.

Le passage de l’approvisionnement synchrone à la production synchrone dans la chaîne logistique

Vincent **Giard**[1] & Gisele **Mendy**[2]

Résumé

Le passage de l’approvisionnement synchrone à la production synchrone constitue une alternative aux configurations logistiques de proximité souvent utilisées dans les chaînes logistiques orientées vers la production de masse de produits fortement diversifiés et devant satisfaire très rapidement les clients, comme c’est le cas dans l’industrie automobile. Cette transformation offre de nouveaux degrés de liberté susceptibles, sous certaines conditions, d’apporter des gains d’efficience. Figer plus tôt certaines décisions accroît le risque de divergence entre prévision et réalisation. De nouveaux mécanismes correcteurs doivent être imaginés. On s’attache ici à identifier ces transformations et leurs conséquences pour les partenaires de la chaîne logistique, en s’appuyant sur le concept de Point de pénétration de commande qui facilite l’analyse de l’interdépendance entre les processus d’une entreprise et ceux de ses fournisseurs.

Mots clefs : chaîne logistique, processus, point de pénétration de commande, différenciation retardée, production synchrone.

Abstract

The transition from synchronous supplies to synchronous production represent a possible alternative to the neighborhood logistics configurations, especially in the automotive industry. It permits to gain, on certain conditions, efficiency and more freedom of manoeuvre faced with a strong variability of the demand on the supply chain. By frozen some decisions earlier, it has for consequence a risk of change between what is expected and what is realized, which is necessary to counter by the implementation of new correctives mecanisms. This article will identify the changes and consequences for partners of the supply chain. The utilization of the concept of Order Penetration Point (OPP) make the analyse of interdependance between client’s process and supplier’s process easier.

Key Words: supply chain, process, Order Penetration Point, postponed customization, synchroneous production.

L’approche de la production à flux tendus suivant les principes du Juste-À-Temps (JAT) s’est diffusée dans les entreprises occidentales dès le début des années quatre-vingt.

[1] Lamsade, université Paris Dauphine, {giard@lamsade.dauphine.fr}
[2] Lamsade, université Paris Dauphine, {gisele.mendy@renault.com}

Au cours de la décennie suivante, les concepts de chrono-compétition et de chaîne logistique se sont progressivement imposés sous la pression de la globalisation. Ils ont déclenché une tension plus grande sur les flux, conduisant au développement de de relations durables entre donneurs d'ordres et fournisseurs, facilitant une proximité géographique plus forte de certains de leurs processus proches de la sortie de produits finis. Ils ont aussi induit une vision plus systémique du réseau de processus sollicités, impliquant la prise en compte simultanée de logiques locales liées à des périmètres juridiques différents et d'une vision globale d'amélioration de l'efficience. Ce changement de perspectives a des impacts sur le JAT et conduit certaines industries, comme celle de l'automobile, à passer progressivement d'une logique d'approvisionnement synchrone à celle de production synchrone, modifiant sensiblement le périmètre spatio-temporel de l'analyse. Cette évolution permet, sous certaines conditions, des gains d'efficience et modifie les degrés de liberté, les contraintes et les risques encourus pour les partenaires de la chaîne logistique. On s'attachera ici à l'analyse de ces transformations et de leurs conséquences sur le plan stratégique, tactique et opérationnel. Le problème du partage de la valeur créée et des risques induits par ces transformations est fondamental parce qu'il conditionne la pérennité de la chaîne logistique et la survie de certains de ses acteurs. Il ne sera cependant pas abordé ici, l'analyse se centrant sur l'examen des transformations potentielles des processus. L'industrie automobile sera prise ici en exemple car c'est la plus avancée dans cette évolution qui concerne les industries orientées vers la production de masse de produits personnalisés (Anderson & Pine, [3], 1997). Les idées développées ici ne relèvent pas d'un retour d'expérience mais de réflexions préalables à une intervention sur le terrain, creusant certaines implications du croisement de concepts bien connus et non mises en évidence à notre connaissance

Dans une première partie, on examinera l'importance de la synchronisation dans la chaîne logistique et on précisera quelques concepts mobilisables dans son analyse. Dans une seconde partie (§ 2, page 267), on analysera les implications du concept de point de pénétration de commande et de son déplacement, conduisant à passer de l'approvisionnement synchrone à la production synchrone.

1 Synchronisation et chaîne logistique

La synchronisation des processus productifs est une préoccupation déjà ancienne (§ 1-1). Elle conduit à adopter une vision plus systémique du réseau des processus sollicités. Cette évolution a un impact sur l'usage des principes du JAT, ce que l'on illustrera au travers d'un exemple industriel automobile avec la mise en place de flux synchrones (§ 1-2, page 261). La mobilisation du concept de pénétration de commande (PPC), facilite l'analyse de l'interdépendance entre les processus d'une entreprise et ceux de ses fournisseurs (§ 1-3, page 264).

1-1 La synchronisation des processus

La chaîne logistique a pour principal intérêt de forcer à une vision processus qui conduit à l'analyse et la résolution de problèmes interdépendants, le plus souvent traités de manière indépendante. On est sur une logique d'intégration et de coordination des activités qui rend plus complexe le pilotage des flux. L'analyse critique des processus s'impose et implique une meilleure visibilité des interactions entre sous-systèmes.

La chaîne logistique peut s'analyser comme un enchaînement de processus de production ou de transport, dans une approche «client-fournisseur». Le processus de production est défini, dans ce cas, avec une granularité telle sa production soit une référence de produit fini ou de composant. Cette relation explicite avec la nomenclature permet de poser le problème de synchronisation des processus, les états intermédiaires d'un produit obtenus à l'intérieur d'un processus ainsi défini ne faisant l'objet d'aucune demande par un autre processus. On commencera par préciser les différences qui existent entre productions à flux tirés et à flux poussés (§ 1-1.1) qui doit tenir compte d'une possible évolution dans le niveau de précision de la demande exprimée (§ 1-1.2, page 260), avant de discuter du concept de synchronisation (§ 1-1.3, page 261).

1-1.1 Productions à flux tirés et flux poussés

La distinction entre production pour stock et production à la commande est ancienne. Le rappel de leurs caractéristiques permet de mieux comprendre la distinction entre production à flux poussés et production à flux tirés, à la base du JAT.

La production à la commande d'un produit fini ou intermédiaire suppose l'existence d'une demande effective d'un client, préalablement à toute mise en production, alors que la production pour stock se fonde sur une demande potentielle, avec ce que cela implique d'incertitude sur les quantités demandées et les moments précis où ces demandes se concrétiseront. L'analyse des gammes et de la nomenclature d'un produit complexe permet d'estimer, pour chaque référence utilisée par ce produit fini, l'intervalle de temps minimal qui sépare le lancement en production de la référence, du moment où s'achève la production du produit fini intégrant cette référence[1]. Si le plus grand de ces intervalles minimaux reste inférieur à l'intervalle de temps séparant la commande du client de sa livraison, la production à la commande du produit fini et de tous ses composants est possible. Dans le cas contraire tout ou partie des composants devra normalement être fabriqué pour stock. Cela étant, la connaissance de la commande n'implique pas nécessairement de produire à la commande, en particulier en cas de production de masse de produits faiblement diversifiés et consommés avec une certaine régularité (petits appareils ménagers, par exemple).

On parle d'assemblage à la commande[2] lorsque, dans le graphe représentant la nomenclature de toutes les références utilisées dans la production du produit fini, la frontière séparant les références produites pour stock de celles produites à la commande est proche de la référence de niveau 0 (produit fini). D'autres raisons peuvent pousser à accroître la part de production sur stock en déplaçant la frontière vers la référence de niveau 0: volonté d'améliorer la réactivité commerciale ou impossibilité d'absorber de fortes variations saisonnières avec les capacités productives installées conduisant à produire pour stock une partie de la demande. Dans ce contexte, en amont de la frontière, la production s'effectue normalement sur la base d'anticipations (flux poussés) et, en aval de la frontière, elle est déclenchée par la demande effective du client du processus concerné (flux tirés). Dans le premier cas, l'ordre de fabrication (OF) est créé par un service de planning sur la base d'un ensemble cohérents de prévisions. Dans le second cas, ce service peut également créer l'OF sur la base de demandes exprimées; dans le JAT, les kanbans jouent le rôle d'ordres de fabrication ouverts.

[1] Selon le contexte, les techniques d'ordonnancement de la série unitaire ou celles de la MRP peuvent être mobilisées pour obtenir cette information. On reviendra sur ce point au § 1-3, page 264 et à la figure 2 de la page 266.

[2] Ce terme d'assemblage à la commande est parfois associé à la production de produits finis fortement diversifiés partant d'un nombre limité d'ensembles faiblement diversifiés de composants ou sous-ensembles plus ou moins interchangeables au montage, conçus dans une approche modulaire (Giard, 2003, [6]).

Pour utiliser le JAT, il faut que les processus productifs de la chaîne logistique allant de ceux qui fabriquent les références normalement produites pour stock, à celui de l'assemblage final, disposent tous de capacités s'adaptant immédiatement aux variations du taux de la demande de la référence concernée. Si tel n'est pas le cas, des désamorçages se produisent nécessairement en l'absence de stock de sécurité dont l'existence est jugée néfaste dans le cadre du JAT. Ceci explique que le JAT ne soit possible que sous la condition stricte d'une variabilité de la demande compatible avec la flexibilité de la capacité mobilisable par les différents processus sollicités.

Pour terminer, il convient de noter que la production à flux tendus n'élimine pas toujours les stocks mais en modifie la localisation (Paché et Sauvage, 1999,[11]). Ceux-ci ont tendance à remonter vers l'amont des processus. Une partie des processus de fabrication s'appuie sur le JAT, tandis que l'autre s'appuie sur des prévisions et marche à flux poussés. Les raisons de cette tendance seront expliquées.

1-1.2 Vers une plus grande précision de la demande

En matière d'approvisionnement externe, l'incertitude génère des coûts additionnels liés à l'existence de stocks de précaution, de ruptures de stock ou à la mobilisation momentanée de ressources pour faire face rapidement à un excédent de charge non prévu. Depuis une vingtaine d'années, les relations entre les grandes entreprises et leurs fournisseurs se sont progressivement transformées par des accords de partenariat permettant de garantir la qualité et de limiter l'incertitude et donc de diminuer les coûts. Ces nouvelles relations font passer d'une situation caractérisée par un passage tardif et brutal d'une absence d'information à une information précise, à une situation dans laquelle le client s'engage progressivement et relativement tôt, vis-à-vis de son fournisseur, sur une demande de plus en plus précise au fur et à mesure que l'on s'approche de la livraison.

Cette évolution de la précision de l'engagement contractuel du client peut prendre deux formes non exclusives:

- Il peut tout d'abord s'agir d'engagement sur des fourchettes relatives au volume de la demande, à sa composition (répartition entre plusieurs références) et aux dates de livraison. Cette focalisation progressive permet au fournisseur de mieux gérer sa capacité et de s'assurer de la disponibilité des composants nécessaires en limitant le risque de stockage excessif.
- Il peut également s'agir de geler d'abord le volume (V), puis la structure (S). L'approvisionnement synchrone (voir § 1-2.1) conduit dans certaines industries de masse produisant des produits fortement diversifiés, comme l'industrie automobile, à demander au fournisseur à livrer un ensemble de composants différents dans un certain ordre (encyclage). Dans ce cas, la dernière incertitude à lever est relative à l'ordre (O). En règle générale, on passe d'une connaissance très grossière sur ces trois composantes (VSO), à état où l'on détermine le volume (VSO), puis la structure (VSO) et, enfin, l'ordre (VSO). Dans certains cas, il faut ajouter la date de livraison comme quatrième niveau d'incertitude; dans les industries de production de masse de produits fortement diversifiés, l'organisation en ligne conduit à une régularité d'approvisionnement qui fait que ces dates ne sont connues que tardivement.

Cette seconde forme de précision succède souvent à la première, lorsque l'on se rapproche de la livraison à effectuer.

Par rapport à ce qui a été dit au § 1-1.1, l'introduction d'un niveau variable de certitude dans l'information communiquée au fournisseur conduit à faire «remonter dans sa nomenclature» la partie de production que l'on peut considérer comme étant à la commande et, corrélativement, à produire pour stock, la partie incertaine faisant l'objet de stocks de sécurité.

1-1.3 Synchronisation

Le concept de synchronisation est habituellement lié à celui de coordination en temps réel du fonctionnement de plusieurs systèmes pour atteindre un objectif de performance de l'ensemble dans de bonnes conditions. La synchronisation ne garantit ni l'efficacité ni l'efficience. Ce sont les procédures mobilisées et le cadre organisationnels qui le permettent. La synchronisation implique seulement que les procédures des systèmes concernés exploitent, en temps réel, certaines informations relatives au pilotage opérationnel de tout ou partie de ces systèmes.

La production à flux tendus répond bien à ce principe si l'on accepte l'idée selon laquelle la coordination en temps réel n'exclut pas que la réaction d'un système aux informations transmises par un autre système puisse avoir des effets différés. Selon le niveau de latence accepté, le périmètre d'application du concept de synchronisation est plus ou moins large. Il semble judicieux d'accepter cette latence dans l'application du concept de synchronisation aux processus productifs, dans la mesure où elle se concrétise par une demande de mise à disposition, à certaines échéances, de quantités de certaines références sans rien stipuler sur la manière d'y parvenir.

La production à flux poussés cherche à assurer la cohérence temporelle des décisions de production prises au niveau des différents processus productifs, sur la base de demandes prévisionnelles. On y retrouve une volonté de synchronisation sans que l'on puisse qualifier facilement ce pilotage des flux, de mécanisme de synchronisation. En effet, la qualité imparfaite des prévisions oblige à mobiliser des mécanismes additionnels d'ajustement pour rendre cohérentes les décisions prises dans les différents systèmes concernés. La remise en cause fréquente et importante de décisions locales initialement prises sur la base d'informations émanant d'autres systèmes rend les mécanismes de rétroaction plus importants que les mécanismes décisionnels initiaux. Il semble préférable de considérer que ces mécanismes correcteurs ne font partie des mécanismes de synchronisation qu'à condition que leur rôle soit mineur afin de conserver aux mécanismes de synchronisation un caractère de décisions prises sur des bases relativement fiables et de ne pas occulter les causes de modifications possibles. Les mécanismes correcteurs visés ici ne concernent pas les décisions exploitant l'amélioration progressive des informations transmises au fournisseur lorsque l'on se rapproche de la livraison, évoquée au § 1-1.2.

On a indiqué en introduction que la tension sur les flux avait conduit à l'approvisionnement synchrone, puis à la production synchrone. Examinons rapidement ce qui caractérise ces formes d'organisation en flux synchrones, sachant que leurs implications seront développées ultérieurement.

1-2 Approvisionnements et flux synchrones

L'allongement du délai laissé au fournisseur transforme les contraintes qu'il subit et donc les réponses qu'il peut mobiliser. Pour l'appréhender, on examinera les caractéristiques de l'approvisionnement synchrone (§ 1-2.1) et celles de la production synchrone (§ 1-2.2, page 263).

1-2.1 L'approvisionnement synchrone

Dans le JAT, les kanbans sont des ordres de fabrication ouverts et interchangeables, la distinction entre kanbans de fabrication et de distribution ne modifiant pas substantiellement ce principe. Le nombre de kanbans est défini par le taux de demande, la taille du lot et le temps mis par un kanban à parcourir le cycle complet le ramenant à son point de départ. Le centre de fabrication de la référence peut être externe, auquel cas on est dans une logique d'approvisionnement. A priori, la synchronisation n'implique rien quant à la longueur du cycle qui peut être ou non très court. Ce n'est donc pas là que se situe la différence.

Dans la pratique, on parle d'approvisionnement synchrone essentiellement dans le contexte de lignes d'assemblage lorsqu'un poste de la ligne est amené à utiliser des composants optionnels et/ou des composants interchangeables (exemple, un moteur choisi parmi trois types possibles): la variété est obtenue par une combinaison de composants choisis dans des ensembles disjoints de composants interchangeables et par la présence ou l'absence de composants optionnels[1]. Dans le premier cas, le fait de ne pas monter systématiquement un composant optionnel sur tous les produits finis conduit à une consommation irrégulière. La gestion par kanban de cette référence oblige à détenir un stock en bord de ligne dont la taille théorique peut correspondre au cumul des lots de tous les kanbans, avec les inconvénients qui en découlent immédiatement lorsque le composant optionnel est encombrant ou onéreux. Le second cas se rencontre essentiellement dans le cadre d'une personnalisation du produit fini dans le cadre d'une approche modulaire. La demande d'un composant donné est également irrégulière et l'on rencontre la même nécessité de créer des stocks en bord de ligne, avec la complication supplémentaire induite par la variété.

Pour limiter ces problèmes, la parade consiste à demander de livrer au début de chaque plage horaire la quantité exacte devant être consommée avant la prochaine livraison; on parle alors d'approvisionnement synchrone. Par rapport à la démarche classique du JAT où l'enlèvement des quantités stockées dans le centre de production se fait au fil de l'eau, on utilise des kanbans de distribution non banalisés, chacun stipulant de livrer une quantité précise à une heure donnée d'un jour donné. Dans le cas de composants interchangeables provenant d'un centre de production, le souci de ne pas multiplier les stocks en bordure de ligne et de limiter le risque d'erreur de sélection de la référence à monter, conduit à demander au fournisseur de configurer sa livraison en panachant les références demandées dans l'ordre correspondant exactement à la consommation sur ligne d'assemblage (encyclage).

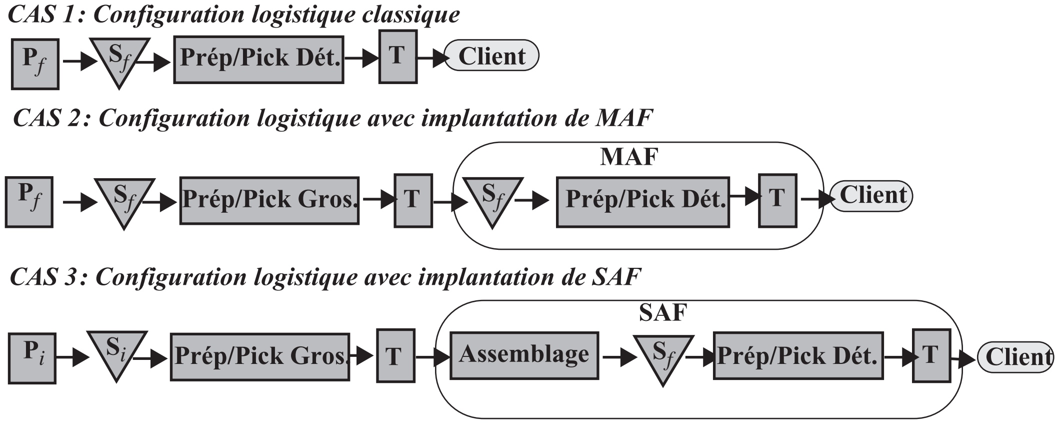
Dans les organisations en ligne, le kanban est émis bien en amont du centre de consommation, compte tenu de la brièveté du temps de cycle de la ligne par rapport au temps nécessaire de préparation, d'acheminement et de mise à disposition. Il dépasse rarement quelques heures, ce qui a conduit à la mise en place d'organisations spécifiques, en particulier dans l'industrie automobile. Le fournisseur est contraint à se rapprocher physiquement du client en raison de la brièveté du temps de réponse imposé. Cette problématique du choix de configurations pour des fournisseurs dans l'industrie automobile a été abordée par Lambert (2002, [5]) dans une optique intégrant la logistique et la production, mettant en évidence les degrés de liberté dont on dispose dans la localisation de certaines opérations de production. La figure 1 s'inscrit dans la continuité de cette réflexion, en schématisant les deux solutions, non exclusives, utilisées:

- Les circuits de distribution physiques peuvent être adaptés par la création de Magasins Avancés Fournisseurs (MAF) localisés à proximité du client. Cette délocalisation des

[1] On est alors en présence de l'une des formes possibles de différenciation retardée, permettant de concilier une certaine rigidité du système productif et sa capacité à répondre à une demande difficilement prévisible et fortement diversifiée, à condition que la structure de nomenclature soit du type «assemblage à la commande» (voir figure 3 de la page 266).

FIGURE 1

Configurations logistiques possibles sur des délais courts d'approvisionnemen


CAS 1 : Configuration logistique classique
P_f
S_f
Prép/Pick Dét.
T
Client
CAS 2 : Configuration logistique avec implantation de MAF
MAF
P_f
S_f
Prép/Pick Gros.
T
S_f
Prép/Pick Dét.
T
Client
CAS 3 : Configuration logistique avec implantation de SAF
SAF
P_i
S_i
Prép/Pick Gros.
T
Assemblage
S_f
Prép/Pick Dét.
T
Client

P$_f$: Production de produits finis   **P**$_f$: Production de produits finis   **T** : Transport
S$_f$: Stockage de produits finis   **S**$_i$: Stockage de produits intermédiaires
Prép/Pick Dét. : Préparation et *picking* détaillé   Prép/Pick Gros. : Préparation et *picking* grossier

stocks de produits finis présente l'avantage de réduire le temps de réaction du fournisseur et l'inconvénient de multiplier ses points de stocks (et donc les stocks de sécurité), si le client a plusieurs sites de production. C'est dans ces MAF que la préparation de la livraison et l'éventuel encyclage sont réalisés.

- On peut également modifier la localisation de certaines opérations de production dans des Sites Avancés Fournisseurs (SAF) qui réalisent, de surcroît les opérations des MAF. Cette solution s'avère nécessaire pour l'approvisionnement synchrone de composants volumineux ou chers à forte différenciation retardée, impliquant un assemblage à la commande (sellerie automobile, par exemple), si l'usine fabriquant ces composants est éloignée du client. Les avantages et inconvénients du MAF sont ceux du SAF, à ceci près que les investissements d'un SAF sont nettement plus importants.

Dans certaines industries, plusieurs fournisseurs d'un même site de client (ou de plusieurs clients proches) peuvent être installés dans une même zone industrielle que l'on qualifie souvent de parcs fournisseurs.

La stratégie de banalisation des sites de production retenue par les industriels depuis quelques années pour améliorer leur flexibilité et réactivité oblige les fournisseurs à multiplier les MAF et SAF, dans des conditions économiques difficilement compatibles avec la réduction des coûts exigées par leurs clients. Le passage de l'approvisionnement synchrone à la production synchrone constitue alors une alternative permettant de concilier, sous certaines conditions, baisse des coûts et efficacité de réponse à la forte variabilité de la demande.

1-2.2 La production synchrone

La volonté de satisfaire rapidement les clients dans un contexte fortement concurrentiel a conduit certaines industries, comme celle de l'automobile, à raccourcir le délai séparant la prise de commande d'un produit personnalisé, de sa livraison au client et à garantir ce délai, alors qu'avant ce délai variait en fonction des possibilités de la production et des approvisionnements. Ce pilotage direct de la production par le carnet de commande pose, pour le fabricant, de redoutables problèmes que l'on n'abordera pas ici.

Le gel de ce qui doit être produit quelques jours à l'avance permet de transmettre au fournisseur les mêmes informations que celles transmises dans le cadre de l'approvisionnement synchrone mais avec quelques dizaines d'heures d'avance au lieu de quelques heures d'avance. Les informations fournies permettent alors au fournisseur d'agir plus en amont dans sa production et de mieux la rationaliser compte tenu de la réduction d'incertitude. On peut parler alors de production synchrone, dans la mesure où les informations partagées par le client et le fournisseur permettent aux deux parties de produire exactement ce qui est demandé. On verra en temps utile que cette transformation s'accompagne d'une modification des risques.

L'usage du concept de point de pénétration de commande permet de mieux comprendre comment le client «rentre» chez le fournisseur, que ce soit en approvisionnement synchrone ou en production synchrone, avec ce que cela implique sur la marge de manœuvre dont il dispose.

1-3 Analyse des mécanismes du point de pénétration de commande sur une chaîne logistique

Dans toute chaîne logistique, un processus de production ou d'assemblage mobilise d'autres processus de production qui fabriquent les composants requis. Lorsque ces processus-amont sont internes à l'entreprise, le pilotage de ces processus est facilité par le fait qu'ils s'inscrivent dans le même périmètre juridique. Il n'en est plus de même lorsque pour les processus amont appartenant à d'autres entités juridiques dont les objectifs, structures et procédures n'ont guère de raisons d'être compatibles avec ceux de leur client. La compréhension des interdépendances entre les processus du client et ceux du fournisseur est facilitée par l'analyse des conséquences du positionnement du point de pénétration de commande (PPC) dans les processus du fournisseur. Après avoir défini dans un premier temps les principes du PPC (§ 1-3.1), on cherchera à mettre en évidence l'impact du PPC, en fonction du mode d'organisation de la production chez le fournisseur (§ 1-3.2, page 265).

1-3.1 Définition du point de pénétration de commande

Ce concept, peu traité dans la littérature, est relativement ancien. Il s'inscrit dans la réflexion conduite sur le *postponement*, en usage dès 1920 et formalisé dans les années cinquante (Alderson, 1950, [2]). Selon ce concept, l'entreprise doit retarder le plus possible l'exécution de certaines opérations de production, d'assemblage ou de conditionnement en ne les déclenchant qu'à la réception de commandes fermes pour répondre, sans stock inutile, aux besoins exacts exprimés par le client. La mise en oeuvre de ce principe sur la chaîne de valeur dépend, en autres, du délai de réponse commercialement admissible séparant une commande ferme, de sa livraison. Si ce délai est court, des stocks de composants et sous-ensembles doivent être constitués en anticipation de la demande. L'augmentation de la diversité des produits et la diminution de ce temps de réponse conduisent à déplacer ces stocks vers l'aval du processus de production.

Le concept de point de pénétration de commande peut également faire référence à celui de «point de découplage» *(CODP, Customer Order Decoupling Point)* introduit par la *Philips Logistics School* aux Pays-Bas et défini comme «*le point sur la chaîne logistique où l'ordre du client pénètre*». Philips est parti du constat que l'ensemble des sous-processus ne pouvant fonctionner à la même vitesse nécessitait l'introduction d'un point de découplage. Biteau *et al*. (1998, [5]) assimilent ce point de découplage à un curseur évoluant sur la profondeur du processus de production et qui fixe le moment où le flux de production cesse d'être piloté sur prévisions et est piloté sur commandes connues. Pour Vallin (2003, [14]), le découplage entre la production à flux tirés et celle à flux poussés est réalisée par un «stock stratégique». Le courant de la chaîne logistique s'intéresse également à ce concept pour ses potentialités d'amélioration de la productivité (Hoover et *al*, 2001, [7]) ou de la flexibilité des systèmes productifs (Roos, 2000, [12]).

Le concept de PPC mobilisé ici reprend l'idée de l'influence, sur l'organisation de la production d'un fournisseur, de l'intervalle du temps séparant l'arrivée d'une commande ferme, du moment où la livraison est requise chez le client. Mais pour séparer sans trop d'ambiguïté les activités réalisées avant l'obtention de la commande ferme, de celles déclenchées consécutivement à cette prise de commande, il faut prendre en compte les gammes et nomenclatures et mobiliser la notion de frontière introduite au § 1-1.1, partageant potentiellement ce qui relève, en amont, d'une production pour stock, de ce qui peut être géré à la commande en aval, mais ne l'est pas nécessairement (on reviendra sur ce point).

En mobilisant le concept d'anticipation de la demande, Tarondeau (1982, [13]) différenciait les systèmes de production par la localisation de stocks de produits ou de biens intermédiaires correspondant à des niveaux d'anticipation différents et impliquant des arbitrages différents entre coûts et risques, dépendant à la fois de la répétitivité de la demande à satisfaire et de l'éloignement du stock par rapport à la dernière étape du processus productif. L'analyse proposée ici apporte des éléments de réflexion additionnels en précisant sous quelles conditions ces stocks peuvent efficacement être introduits et en mettant en évidence d'autres marges de manœuvre.

La figure 2 de la page 266 synthétise la démarche d'analyse combinant les informations relatives au délai dont dispose le fournisseur et aux nomenclatures et gammes concernées, via un graphique Gantt. Ce graphique, correspondant à l'ordonnancement de l'exécution de la commande du client, est établi au plus tard et ne tient compte que des durées et des contraintes d'antériorité (il est donc similaire à celui habituellement utilisé en ordonnancement de la série unitaire). Conformément à la définition retenue pour les processus, chaque activité de ce graphique Gantt correspond au processus de production d'une référence. Les contraintes d'antériorité entre activités sont repérées par les traits verticaux. L'établissement de ce graphique dépend de l'organisation de la production (ligne ou atelier), comme on l'expliquera ultérieurement. Les opérations débutant postérieurement à la date du PPC sont grisées et peuvent être pilotées par les informations de la commande ferme reçue; celles qui débutent avant reposent nécessairement sur des informations prévisionnelles ou de précision moindre comme cela a été analysé au § 1-1.2.

1-3.2 Le point de pénétration de commande et les différents modes d'organisation de la production

Comme on l'a déjà indiqué (§ 1-1.1), on peut avoir intérêt à produire pour stock en cas de production de masse de produits faiblement diversifiés et consommés avec une certaine régularité. Dans ce cas, le fournisseur n'utilise pas les informations du PPC qui sont sans réelle valeur ajoutée. L'approvisionnement de bougies dans une usine d'assemblage automobile est un bon exemple de ce cas de figure. Pour préciser ce point, il faut revenir sur la relation existant (Giard, 2003, [6]) entre la nomenclature de production, la variété des produits et la variété des composants en regardant la forme des nomenclatures habituellement rencontrées dans les trois modes de production illustrée à la figure 3 de la page 266.

FIGURE 2
Le principe du point de pénétration de commande (PPC)

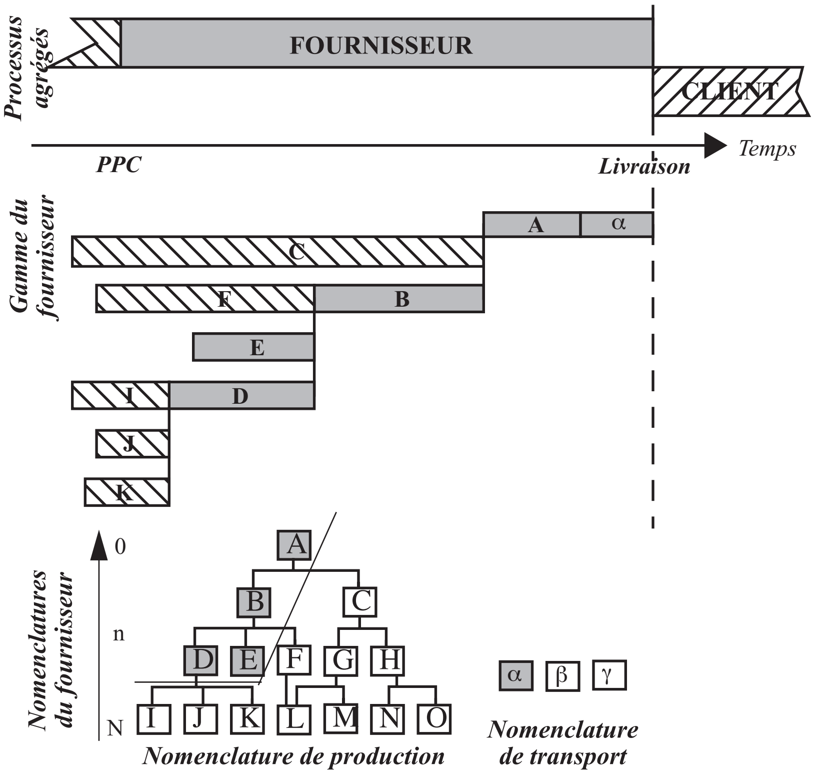
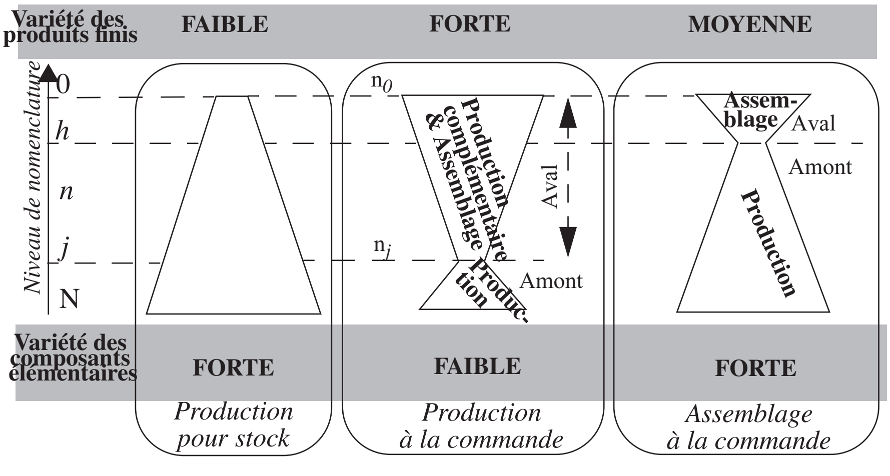


FIGURE 3
Nomenclature de production et organisation de la production en fonction du marché



Le premier mode de production correspond à l'exemple présenté au paragraphe précédent. Dans le second cas de figure — production à la commande — les entreprises produisent des composants de faible diversité rentrant dans la majorité des produits finis (au-dessus du goulet d'étranglement de la nomenclature) et procèdent ensuite à la production et assemblage de composants spécifiques. On notera *j* ce niveau de nomenclature à partir duquel on commence à introduire la diversité des produits intermédiaires et gammes de production spécifiques jusqu'à obtention d'une variété suffisante de produits finis. Dans la production à la commande, la connaissance précise de la commande à honorer, permettra de gérer d'autant mieux la production que le PPC sera remonté suffisamment dans le temps car on a intérêt à ne pas stocker de composants créant de la diversité, (ceux situés au-dessus du goulet d'étranglement). D'une manière générale, pour un goulet d'étranglement situé à un niveau *k*, plus le rapport entre n_0 et n_k (nombre de références nécessaires à un niveau de nomenclature *0* et *k*) est grand et plus le PPC s'éloigne du niveau 0, sans dépasser le niveau *k*, plus forte seront les économies potentielles réalisées sur les stocks (on reviendra sur ce point); en dessous du «goulet d'étranglement» les bénéfices escomptés seront faibles ou nuls.

Dans le troisième cas de figure, les entreprises fabriquent une large gamme de produits finis standardisés dont la variété s'appuie sur la combinaison d'options permettant de personnaliser chaque produit en fonction de la demande. La variété va s'appuyer sur une opération d'assemblage à la commande à un niveau de nomenclature que l'on note *h* et qui repose sur une conception modulaire de certains sous-ensembles associée à une différenciation du produit final par le client. On peut noter ici que plus le goulet d'étranglement de la nomenclature est élevé dans les deux schémas de la figure 3, plus la mise en œuvre de la différenciation retardée est forte. Dans ce cas de figure, si le PPC se situe en dessous du goulet d'étranglement la valeur ajoutée de cette information portera davantage sur les gammes de production, la diversité jouant au-dessus du goulet d'étranglement. Nous reviendrons sur ce point au § 2-1.1, pour dissocier l'effet stock, de l'effet nomenclature.

D'une manière générale, la valeur ajoutée que peut tirer le fournisseur d'une information transmise plus tôt dans ses processus de production dépend du mode de production qu'il utilise (production pour stock et *picking* à la commande, production pour stock et assemblage à la commande, ou production à la commande) et du positionnement du PPC par rapport au goulet d'étranglement représentant l'introduction du niveau de diversité de chaque structure correspondante.

2 Les implications du positionnement du point de pénétration de commande et de son déplacement

Dans cette partie, on suppose que le type de production justifie l'usage d'une production à la commande en aval du PPC, dans une approche d'approvisionnement synchrone ou de production synchrone. L'organisation de cette production influe sur le périmètre d'activités sur lesquelles on peut agir (§ 2-1). Le déplacement de ce PPC sera l'occasion de montrer la variation de degrés de liberté dont les fournisseurs disposent qui peuvent conduire à modifier les décisions stratégiques, tactiques et opérationnelles (§ 2-2, page 271). Ce déplacement implique également une transformation des risques et de leur gestion (§ 2-3, page 272).

2-1 Mode de production et point de pénétration de commande

Pour un fournisseur, l'impact de la connaissance qu'il a du PPC n'est pas le même selon son mode de production. On commencera par analyser le cas le plus simple, celui d'une organisation en ligne de production (§ 2-1.1), avant d'aborder celui d'une organisation en *flow shop* ou en *job shop* (§ 2-1.2, page 270).

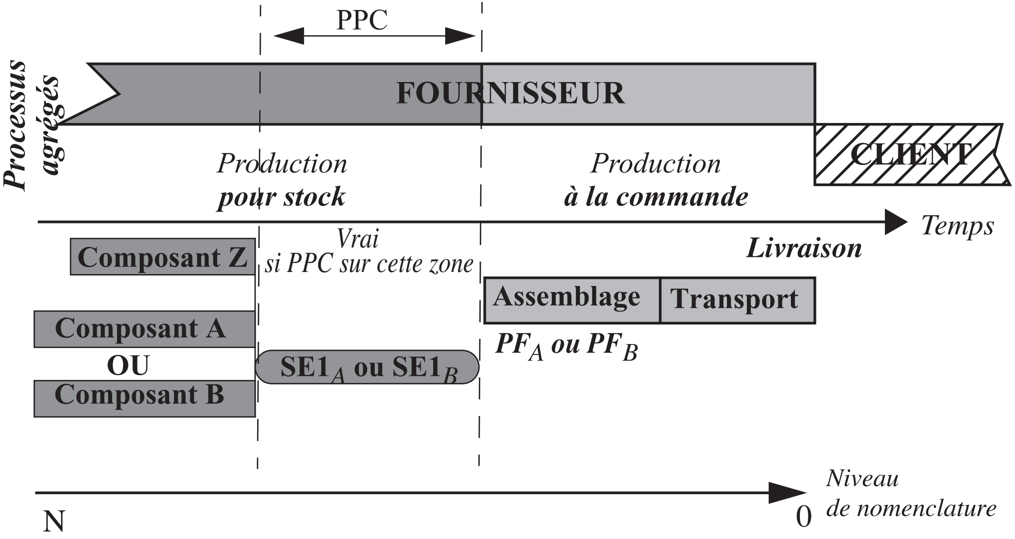
2-1.1 Processus du fournisseur organisé en ligne

Une ligne de production se caractérise par un ensemble d'équipements agencés pour permettre à un flux de transiter systématiquement par la même séquence de postes de travail, en utilisant des moyens automatisés de transfert d'un poste au suivant. La régularité du débit de la ligne fait que les opérations se succèdent sans temps mort. Pour pouvoir appliquer le raisonnement conduit sur la figure 2 de la page 266, il faut que chaque activité conduisant à la production d'une référence soit exécutée sur une ligne dédiée, ce qui conduit à une structure arborescente de lignes de fabrication ou d'assemblage. Ce cas, sans doute peu fréquent, permet d'amorcer le raisonnement avant d'aborder le cas plus compliqué d'organisations productives en *flow shop* ou en ateliers spécialisés, obligeant à raisonner en univers aléatoire.

La durée cumulée de l'ensemble des opérations réalisée sur une ligne, correspondant ici à une activité, est souvent importante. Dans le cas d'un approvisionnement synchrone, peu de ces activités ont des chances d'être pilotées «à la commande». L'éloignement de la frontière du PPC lié au passage à la production synchrone augmente notablement le nombre d'activités éligibles à cette forme de pilotage.

Examinons l'impact potentiel du déplacement du PPC sur le niveau des stocks de composants et sous-ensembles. La figure 4 de la page 268 décrit par un Gantt fléché l'enchaînement des processus de production. L'opération de transport porte généralement sur un lot de produits finis. Chaque activité est donc décrite par un trait dont la longueur est proportionnelle au temps nécessaire pour la fabrication des composants requis par ce lot, en tenant compte du fait que la nomenclature peut impliquer que le produit fini utilise plusieurs unités du même composant.

FIGURE 4
Positions du PPC et Effet stock

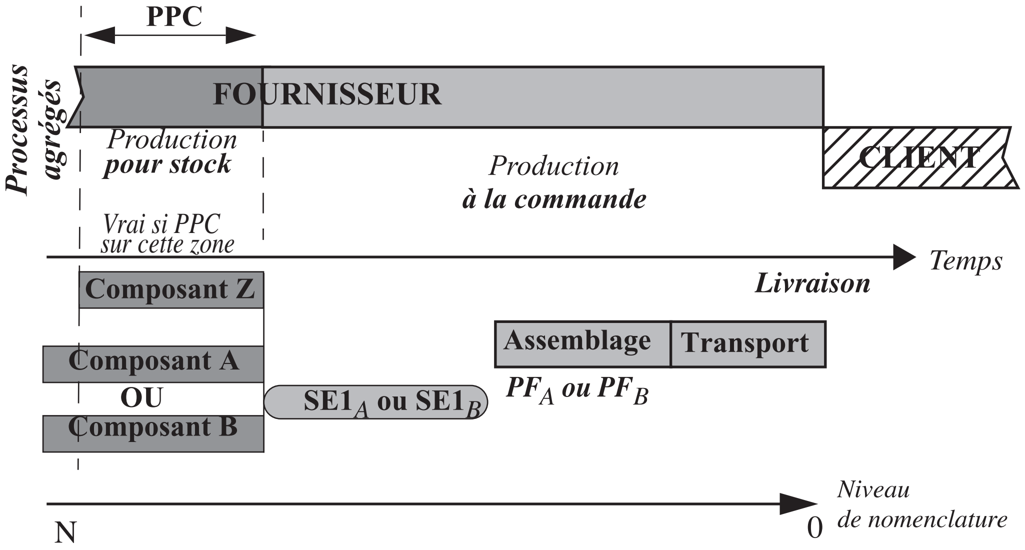


Dans cet exemple, le fournisseur procède à la production pour stock de composants standards Z et de composants optionnels mais non facultatifs A ou B (l'un de ces deux composants étant nécessairement retenu). La seconde étape de son processus de production consiste à fabriquer des sous-ensembles SE$_A$ (utilisant le composant A) ou SE$_B$ (utilisant le composant B). En amont de la frontière du PPC, le fournisseur doit produire pour stock, avec une prise en compte de l'incertitude par des stocks de sécurité (et donc ici pour les sous-ensembles SE$_A$ ou SE$_B$). En aval de la frontière du PPC, le fournisseur peut procéder à une opération d'assemblage à la commande, suivie d'une opération de livraison vers l'usine cliente, sans constitution de stock de sécurité.

La remontée dans le temps du PPC génère, pour le fournisseur, de nouvelles marges de manœuvre, ce qu'illustre la figure 5. L'incertitude sur le volume et la structure des sous-ensembles SE$_A$ ou SE$_B$ disparaît, permettant au fournisseur de supprimer les stocks de sécurité de ces références, diminuant mécaniquement les coûts d'immobilisations. Ce raisonnement se généralise sans difficulté pour plusieurs références dont les activités de productions franchissent la frontière du PPC.

FIGURE 5

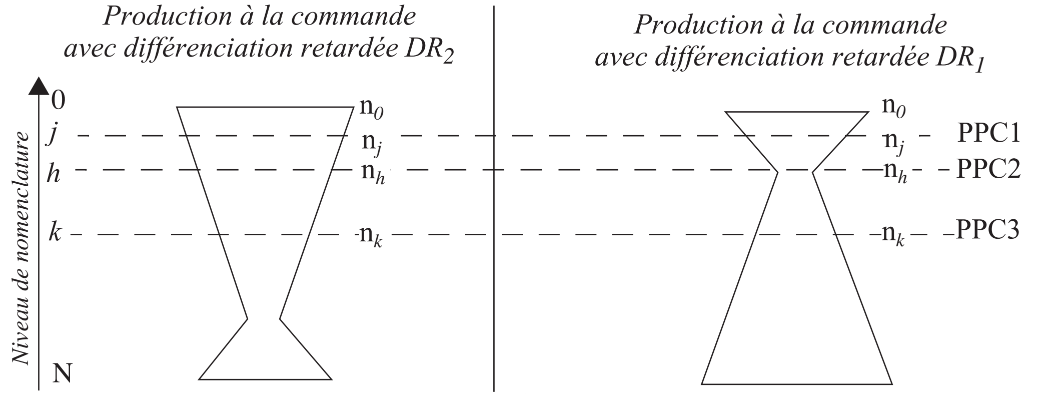
Déplacement du PPC et Effet stock



Si l'on reprend notre réflexion faite sur la relation existant entre la valeur ajoutée du positionnement d'un PPC et le mode de production du fournisseur (§ 1-3.2, page 265), on peut mettre en évidence que les gains potentiels pour le fournisseur varient en fonction du niveau différenciation retardée retenu. La figure 6 de la page 270 permet de comprendre le mécanisme impliqué.

Notons n_0, n_1, …, n_N le nombre de références des niveaux de nomenclature 0, 1, …,N. Pour simplifier, on considérera que, du niveau 0 au niveau *j* (ou *h* ou *k*) de la nomenclature, tous les composants peuvent être fabriqués à la commande, les autres étant fabriqués pour stock. On distingue ici deux situations, celle d'une différenciation retardée forte, notée DR$_1$, et celle d'une différenciation retardée faible, notée DR$_2$. Dans le premier cas, le rapport n_j / n_0 est plus fort que dans le second cas, ce qui implique qu'à diversité identique de produits finis (même valeur de n_0 dans les deux cas), on observe une variété de composants fabriqués pour stock plus forte dans le cas DR$_2$ que dans le cas DR$_1$. Cette diversité s'analysant à partir de l'ensemble des possibles doit être adaptée à une demande précise car elle ne mobilise qu'une partie de la variété potentielle ($n'_0 < n_0$). On peut

FIGURE 6
Production à la commande avec ou sans Différenciation retardée et identification de marges de manœuvre pour le fournisseur


Production à la commande avec différenciation retardée DR_2
Production à la commande avec différenciation retardée DR_1
Niveau de nomenclature
0
j
h
k
N
n_0
n_j
n_h
n_k
n_0
n_j
n_h
n_k
PPC1
PPC2
PPC3

raisonnablement penser que le rapport n'_j/n'_0 reste plus fort pour DR$_2$ que pour DR$_1$. pour des utilisations similaires de cette diversité finale.

La remontée du point de pénétration de commande conduit donc à remplacer relativement plus de productions pour stock par des productions à la commande dans le cas DR$_1$ que dans celui de DR$_2$. Ceci conduit à des économies potentiellement plus importantes parce qu'avec le déplacement de frontière, on n'a plus à produire certains sous-ensembles dont on sait maintenant qu'ils ne seront pas utilisés prochainement et parce que les stocks de sécurité des sous-ensembles restants sont devenus inutiles. Dans le cas DR$_1$, le déplacement du point de pénétration de commande en dessous du niveau *h* (correspondant aux composants de base utilisés pour créer la diversité désirée dans un processus d'assemblage) ne présente d'intérêt qu'au niveau des stocks de sécurité, toutes les références restant à produire.

2-1.2 Processus du fournisseur organisé en *flow shop ou* en *job shop*

Si les processus de production du fournisseur utilisés en aval du PPC ne sont pas organisés en ligne de production ou d'assemblage, deux phénomènes viennent compliquer l'analyse.

- L'existence de temps et coûts de lancement incite généralement à travailler par lots, au terme d'un arbitrage entre des coûts de lancement et de possession, tenant compte des contraintes du transport final (taille des lots transportés et délai de mise à disposition pour les références de niveau supérieur à 0). La durée d'une activité sur le Gantt de la figure 2 (page 266) intègre alors un temps de lancement et le temps total de fabrication du lot.
- L'organisation en ateliers implique la résolution périodique d'un problème d'ordonnancement. Ce problème est plus facile à résoudre dans le cas d'une structure en *flow shop*[1] que dans le cas d'une structure en *job shop*[2]. Dans un cas comme dans l'autre,

[1] Ce type de production se distingue de celui d'une organisation en ligne par trois caractéristiques (Giard, 2003, [6]):
- une tâche peut ne pas faire appel à tous les centres de production (l'ordre d'exécution reste le même),
- une dispersion importante des temps opératoires des opérations exécutées sur un même poste de travail,
- l'existence possible de files d'attente, de longueur variable dans le temps, en amont des différents postes de travail.

[2] Ce type de production se distingue de celui d'une organisation en *flow shop* par deux caractéristiques (Giard, 2003, [6]):
- la coexistence de très nombreux cheminements de flux de production dans un même système productif,
- les opérations des tâches à exécuter sur un ensemble de centres de production différents peuvent l'être dans un ordre quelconque.

s'ajoute au temps décrit au paragraphe précédent un temps d'attente qui dépend du portefeuille d'ordres de fabrication en cours, des principes d'ordonnancement retenus et de la configuration productive. Au moment où la commande ferme arrive, il est *a priori* possible de déterminer ce temps d'attente mais, d'une commande à l'autre, ce temps d'attente varie. S'il est possible d'obtenir une estimation crédible de la distribution de ces temps d'attente, l'examen de l'impact du PPC passe par l'analyse d'un problème stochastique d'ordonnancement de la série unitaire pour lequel des démarches simulatoires de type Monte Carlo (Giard, 2003, [6]) permettent de déterminer la probabilité qu'une activité a d'être d'un côté ou de l'autre du PPC. Ce déplacement du PPC peut conduire le fournisseur à adopter de nouvelles règles de lotissement exploitant les nouveaux degrés de liberté, pour baisser les coûts.

2-2 Conséquences pour le fournisseur du déplacement du point de pénétration de commande

Le gel de ce qui peut être produit à l'avance va permettre d'introduire de nouvelles marges de manœuvre au sein des décisions de gestion des niveaux stratégiques, tactiques et opérationnels, pour reprendre la typologie introduite par Anthony (1965, [4]):

- *Parmi les décisions stratégiques*, les fournisseurs ont la possibilité d'envisager une solution alternative aux SAF et MAF par une délocalisation physique de l'outil productif. Il s'agit dans ce cas d'envisager, par exemple, la délocalisation de la production de composants de pièces dans des pays économiquement plus attrayants. Il peut s'agir également de ne délocaliser que certaines opérations tout en conservant pour les magasins de proximité ou sites avancés certaines nouvelles fonctionnalités (zones de ré-encyclage en fonction des aléas rencontrés…). La formulation de la politique de localisation de l'outil productif des fournisseurs se traduit par une transformation des gammes et nomenclatures exploitant les nouvelles marges de manœuvre pour améliorer l'efficience des processus.

  La localisation physique de l'outil productif des fournisseurs, ses caractéristiques (capacité, références, gammes…) peut prendre en compte, dans le cadre d'une production synchrone, la possibilité de déterminer une localisation pertinente en recherchant le barycentre des différents points de livraisons pour plusieurs clients. On est sur des logiques de maillage permettant d'envisager des parcs fournisseurs communs multi-clients permettant de larges économies d'échelle.
- *Les décisions tactiques* s'inscrivent dans un cadre contraint par les décisions stratégiques et peuvent être modifiées sur certains aspects:
  - La planification de la production peut être adaptée pour tenir compte du déplacement de la frontière production pour stock / production à la commande.
  - La définition et préparation *du plan Transport* en univers certain se trouvent également modifiées par le déplacement du PPC avec une localisation et des règles locales qui ne sont plus les mêmes. On a vu précédemment que le déplacement du PPC se caractérise par une certaine marge de manœuvre dans la localisation des opérations de production, ce qui a une incidence non seulement sur la fréquence des livraisons (réduction possible du nombre de fréquences) mais également sur le volume à acheminer (meilleure optimisation des taux de remplissage) et sur la capacité optimale de moyens de transport (sachant que dans l'industrie automobile de gros efforts ont déjà été réalisés).

    Une bonne organisation de plan transport est souvent un enjeu majeur en terme de coûts et de délais. Mais dans le cadre de la production synchrone, le transport de masse ne répond pas au mieux aux nouvelles attentes : de nouvelles formes organisationnelles assurant flexibilité et réactivité doivent être imaginées. C'est dans cette perspective que certains constructeurs automobiles organisent des tournées multifournisseurs à fréquence multi-quotidiennes ; cette décision tactique vise, en fonction de nouvelles caractéristiques globales de flux, à déterminer une gamme optimale de transport où chaque acheminement sélectionne les points d'enlèvement (en fonction de leur situation géographique, contraintes de production…), définit l'ordre de passage et les tailles de lots et tient compte des aléas transport. Ce schéma très utilisé chez les constructeurs Japonais (plus connu sous le terme de « Heijunka » qui signifie « petite quantité, grande fréquence ») permet de s'inscrire dans une logique de localisation des stocks de sécurité dans les véhicules (circulants ou en attente de déchargement sur le parking client) et assure également une meilleure gestion du risque transport (lots répartis dans plusieurs véhicules).

- *Les décisions opérationnelles* : elles consistent à assurer davantage la flexibilité quotidienne et la réactivité nécessaire pour faire face aux aléas (mode correctif) dans le respect des décisions tactiques. Nous reviendrons dessus plus en détail au § 2-3.2.

L'amélioration des processus repose sur leur modélisation. Leur simulation en univers certain permet de mieux cerner les gains d'efficience que l'on peut espérer. Ces gains seront atténués par l'impact de perturbations que des mécanismes correcteurs pourront contrer à condition d'en accepter le coût (cf. § 2-3, page 272). Seule une approche complémentaire de simulation stochastique est alors en mesure de vérifier la pertinence et l'efficacité des mesures correctives proposées.

2-3 Modification des risques induite par le déplacement vers l'amont du point de pénétration de commande

Le déplacement de la frontière du PPC offre de nouvelles marges de manœuvre permettant d'accroître l'efficience des processus. En figeant plus tôt certaines décisions, on augmente le risque de divergence entre ce qui est prévu et ce qui se réaliserait en l'absence d'intervention. Ces gains potentiels d'efficience ont pour contrepartie une entropie accrue qu'il faut contrer par de nouveaux mécanismes correcteurs. L'identification préalable des sources de perturbation possibles des processus concernés par la production synchrone (§ 2.3.1) permet de repérer quelques mécanismes correcteurs possibles (§ 2.3.2).

2-3.1 Identification des perturbations possibles

Dans les systèmes productifs d'inévitables incidents font que tout ne se passe pas comme prévu chez le client comme chez le fournisseur, avec comme conséquence possible une désynchronisation des flux que l'on peut espérer éviter par des mesures correctives. Cette désynchronisation se traduit par des rendez-vous manqués entre des références approvisionnées et des produits en cours d'assemblage chez le client. L'analyse qui suit est fortement marquée par l'exemple de l'industrie automobile.

Le client produisant sur ligne de fabrication ou d'assemblage peut désynchroniser les flux en modifiant la vitesse de son flux ou en modifiant l'ordre des produits personnalisés dans le flux.

L'indisponibilité momentanée de certaines ressources (machines en panne, approvisionnements en retard, renforts indisponibles pour traiter des surcharges de travail ponctuelles sur certains postes d'une ligne…) peut conduire à un arrêt momentané de travail sur un poste, se propageant progressivement sur les postes suivants. Ce désamorçage de la ligne peut être retardé par l'existence de stocks-tampons et le retard pris peut être rattrapé ultérieurement par un allongement de la durée de travail. En tout état de cause, sur certains postes et pendant un certain temps, des produits arriveront en retard et les approvisionnements requis, arrivés comme prévu, s'accumuleront chez le client en bord de ligne ou dans des aires de stockage dédiées.

Le traitement en ligne de la qualité peut conduire à retirer momentanément un produit de la ligne, pour le réintroduire quelque temps plus tard, après rectification des défauts constatés. L'absence d'un composant peut éventuellement induire une décision similaire, si le désamorçage est refusé en raison de son importance et à condition que cette dérivation soit physiquement possible. Dans un cas comme dans l'autre, le produit aura perdu quelques rangs dans la séquence initiale du lancement en production, tandis que ses successeurs en auront gagné un, sauf à connaître à leur tour un problème de qualité (ou de rupture d'approvisionnement) conduisant à leur retrait momentané de la ligne[1]. Cette ***perturbation de l'ordre*** peut, sous certaines conditions, être corrigée par le passage ultérieur des produits dans un *stock de tri* utilisant des règles de sortie autres que celle du «premier entré - premier sorti». Ce rattrapage n'est possible qu'à condition que la capacité du stock de tri soit au moins égale au nombre maximal de rangs perdus par un produit à la suite d'une rectification de défaut (ou une rupture d'approvisionnement). Cette condition n'est pas toujours facile à respecter, aussi certains produits rateront-ils le rendez-vous prévu sur certains postes recevant des composants spécifiques, en arrivant en avance, ce qui peut induire un désamorçage, ou en retard, ce qui provoque un stockage des composants en attente.

Ce changement d'ordre peut, à son tour, conduire à un désamorçage sur d'autres postes. En effet, l'ordonnancement initial tient compte de la variabilité du travail effectué sur certains postes de la ligne en fonction de la présence de composants optionnels à monter[2]. Sur ces postes, un véhicule doté d'une option impliquant une charge de travail supérieure au temps de cycle de la ligne, doit être suivi de plusieurs véhicules ayant une charge de travail inférieure au temps de cycle, pour permettre de rattraper le temps perdu. Sur quelques postes, cette contrainte d'espacement est intangible en raison d'une contrainte capacitaire d'équipement et la violation de la contrainte d'espacement conduit automatiquement à un désamorçage. Sur les autres postes, le désamorçage peut être évité en appelant en renfort du personnel et la synchronisation des flux n'est alors pas remise en cause.

La désynchronisation des flux peut être également imputable au fournisseur pour des raisons de production mais aussi en raison de problèmes d'acheminement.

- Il est normal de retrouver chez le fournisseur organisé en ligne de production les sources de perturbation évoquées chez le client. Certaines d'entre elles disparaissent si la diversité de la production n'implique pas, sur certains postes, de charge de travail variant en fonction des caractéristiques du produit traité. Si les processus concernés s'appuient sur des organisations en *flow shop* ou *job shop*, une attention plus grande doit être portée à l'ordonnancement et aux procédures de réactivité aux incidents. La synchronisation des flux implique le respect d'un encyclage précis à la livraison mais cet ordre ne s'impose pas dans le pilotage des processus en amont de la phase finale de la préparation de la commande; dans ce cas, les contraintes pesant sur ces processus

[1] Une analyse détaillée de ce mécanisme peut être trouvée dans Giard (2003, [6]), p. 603-609.

[2] Voir Giard (2003, [6]), p. 614-619.

sont moins strictes. Si la diversité des produits est forte, le processus qui gère cette différenciation peut être plus ou moins tardif, ce qui conduit à un intérêt plus ou moins fort du gel précoce de la demande du client, comme on l'a déjà évoqué. Une différenciation fortement retardée minimise l'impact d'aléas survenant sur partie de processus produisant pour stock. Un retard de production chez un fournisseur aura des répercussions sur le transport qui constitue le dernier maillon de la chaîne.

- Le problème transport auquel se trouve confronté le fournisseur est celui de la gestion des lots transportés, un retard en production pouvant conduire à faire partir des lots incomplets et à organiser des transports additionnels. Le temps de transport peut être affecté par divers aléas qui conduisent à se prémunir de différentes façons (délai de sécurité…). Enfin, le fournisseur doit réagir à des retards de production pris par le client. Dans certains cas, il devra également être en mesure de modifier l'encyclage si le client l'exige. Ces modifications de dernière minute peuvent être difficiles à intégrer.

Si un certain niveau d'entropie est inévitable, tant chez le client que chez le fournisseur, et doit progressivement baisser par l'appel à des démarches d'amélioration continue, des mécanismes correcteurs sont indispensables pour s'assurer que les rendez-vous pris soient respectés.

2-3.2 Mécanismes correcteurs possibles

Les mécanismes mis en jeu relèvent des décisions opérationnelles qui vont assurer la flexibilité quotidienne nécessaire pour faire face aux aléas. Plusieurs pistes de solutions sont envisageables chez le client et le fournisseur.

On a vu que chez le client les perturbations concernent la vitesse du flux et l'ordre des produits. Sur le premier point, le rattrapage ne peut se faire que par un allongement momentané de la durée du travail. Le retard pris par le flux du client par rapport aux flux des fournisseurs se traduit par la création de stock qui peuvent poser des problèmes matériels importants et pousser le client à demander à ses fournisseurs de ralentir leurs flux pour assurer la synchronisation sur les postes d'assemblage. Le second type de perturbations est plus difficile à gérer. Plusieurs mécanismes sont, a priori, envisageables :

- Il convient d'abord d'attaquer le problème à sa racine en améliorant les processus de production mais aussi ceux d'élimination des défauts rencontrés. Il ne s'agit alors pas de mécanismes correcteurs mais de décisions tactiques, voire stratégiques. On a évoqué déjà la possibilité de restaurer un ordre perturbé par l'intermédiaire de stocks de tri ; l'amélioration des algorithmes de sélection du produit à sortir de tels stocks peut limiter plus ou moins l'impact des perturbations d'ordre. On peut également décider de multiplier de tels stocks de tri, en en plaçant un après chaque processus pouvant conduire à de telles perturbations d'ordre. Ces décisions ne relèvent pas non plus du niveau opérationnel.
- L'élimination totale de ce type de perturbations étant peu vraisemblable, il faut alors imaginer d'autres solutions.
  - La prise en compte des décyclages peut se faire par des stocks de sécurité (ou stock de substitution) en bord de ligne qui dépend avant tout de l'importance des perturbations de l'ordonnancement initial. En cas de retard pris par le client, les composants livrés peuvent y être momentanément entreposés. Cette solution est difficile à utiliser en cas de composants encombrants ou de forte diversité en raison des manipulations rendues nécessaires. On peut également envisager la mise en place de zones de ré-encyclage déportées par rapport à la ligne.

- La technique du baptême progressif est également envisageable. Il s'agit d'une approche fondée sur le principe de la différenciation retardée dans laquelle on affecte au dernier moment au composant le numéro du produit sur lequel il sera monté. Ceci permet, par rapport à une affectation précoce, de permuter deux composants identiques devant être montés sur des produits différents dont l'ordre d'arrivée sur le poste de travail a été permuté à la suite d'aléas.

Pour le fournisseur, on retrouve des mécanismes similaires en ce qui concerne les aléas de production. Il doit en outre prévoir des stocks de sécurité dans son site productif pour être en mesure de respecter ses engagements. Il peut enfin mobiliser des ressources additionnelles (heures supplémentaires…) pour rattraper certaines perturbations. Les mécanismes de réactivité aux aléas de transport sont de trois ordres:

- On peut imaginer de mettre en place des stocks de sécurité en usine. Dans le cadre de production synchrone, il est habituel de prendre en compte la possibilité d'une livraison retardée en introduisant dans les gammes transport un temps additionnel de sécurité en usine avant livraison bord de chaîne. Pour une consommation en bord de chaîne prévue pour la pièce A à un instant *t*, le véhicule doit arriver à *t–x*, *x* dépendant de la distance à parcourir et du type de flux (flux direct, tournée collectage…). Ce stock de sécurité encyclé peut être physiquement déchargé dès réception en usine ou bien être positionné sur un parking de remorques prévu à cet effet.
- On peut aussi déclencher une livraison spéciale. Le système d'urgences utilisé dans l'industrie automobile est plus complexe ici car le fournisseur est amené à reproduire un certain nombre de références qui doivent être également encyclées. Selon son positionnement géographique, son mode de production, le temps de réaction sera plus ou moins grand.
- Le fournisseur peut enfin adapter les caractéristiques des lots transportés, en particulier en procédant à un fractionnement pour éviter tout désamorçage chez son client: afin de contrer les arrêts de chaîne en usine.

Cette liste sans doute non exhaustive de mécanismes correcteurs possibles permet de construire des scénarios qui peuvent être testés dans le modèle de simulation décrivant le fonctionnement de la partie de la chaîne logistique concernée par l'approvisionnement synchrone. Cette simulation permet d'évaluer la robustesse et la pertinence d'alternatives décisionnelles envisagées, à condition que la représentation simplifiée des processus soit pertinente et que les lois utilisées pour simuler les incidents soient réalistes.

3 Conclusion

La mise en œuvre de la production synchrone par l'intangibilité de la commande passée au fournisseur n'est pas sans rappeler celle du JAT. Le mot d'ordre du respect de la séquence d'une production diversifiée a les mêmes vertus que celui du «zéro stock». Pour le JAT, les défauts, les rebuts, les retouches, les pannes de machine provoquaient des perturbations dont la propagation est freinée par les stocks constitués entre processus. Diminuer volontairement ces stocks contraint à s'attaquer aux causes de ces dysfonctionnements (métaphore bien connue de la rivière[1]). Cette chasse aux dysfonctionnements

[1] Selon l'image de Taiichi Ohno, père de la philosophie JAT chez Toyota, on peut comparer les stocks au niveau de l'eau dans une rivière. Dans l'approche traditionnelle, les responsables considèrent que plus le niveau de l'eau est élevé, plus la navigation est aisée car cela permet de s'affranchir des risques que représentent les récifs. Ohno propose au contraire de faire baisser le niveau d'eau pour laisser apparaître les récifs et les éliminer.

conduisant au non respect des rendez-vous entraîne une amélioration des processus sur lesquels l'entreprise fonde sa performance du long terme. Ce nouveau mot d'ordre du respect du séquencement repose implicitement sur la même démarche.

4 Bibliographie

[1] B. Agard, *Contribution à une methodologie de conception de produits à forte diversité*, thèse de doctorat en génie industriel de l'INPG, 2002.

[2] W. Alderson, *Marketing Efficiency and the Principle of Postponement,* Cost and Profit Outlook, 3, Septembre 1950.

[3] [2] D. M. Anderson, J. Pine II, *Agile Product Development for Mass Customization: How to Develop and Deliver Products for Mass Customization, Niche Markets, JIT, Build-to-Order and Flexible Manufacturing*, McGraw-Hill, 1997.

[4] R. N Anthony, *Planning and Control Systems : a framework for analysis*, Harvard University Press, 1965.

[5] R. & S. Biteau. *Maîtriser les flux industriels : les outils d'analyse*, Éditions d'organisation, Paris, 1998.

[6] V. Giard. *Gestion de la production et des flux*, Economica, Paris, 2003.

[7] W. Hoover, E. Eloranta, Jan Holmström et K. Huttunen, *Managing the demand supply chain : Value Innovations for customer satisfaction*, Wiley Operations Management, New York, 2001.

[8] R. Lambert, *Modélisation des choix de configurations logistiques : les leviers d'actions face à la réduction du délai d'approvisionnement constructeurs dans l'automobile*, Thèse de Doctorat, Université du Havre, 2002.

[9] H. L. Lee, «Effective inventory and service management through product and process redesign», *Operations Research*, vol. 44, p. 151-159, 1996.

[10] H.L. Lee et C.S. Tang, «Modelling the costs and benefits of delayed product differentiation», *Management Science*, vol. 43, n° 1, p. 40-53, 1997.

[11] G. Paché et T. Sauvage, *La logistique : enjeux stratégiques*, Editions Vuibert, p. 47-51, Paris, 1999.

[12] H. B Roos, *The concept of the Customer Order Commercial Decoupling (CODP) in logistics management ; a case study approach*, Rotterdamn, May 2000, http://www.few.eur.nl/few/people/roos.

[13] JC Tarondeau, *Produits et technologies, Choix politiques de l'entreprise industrielle*, Collection Dalloz Gestion, 1982.

[14] P. Vallin, *La logistique – modèles et méthodes du pilotage des flux*, Economica, Paris, 2003.

[15] RI Van Hoek, HR Commandeur et B Vos, *Reconfiguring logistics systems through postponement strategies,* Journal of Business logistics, vol.XIX, n˚ 1, 1998.

a Hierarchical Database Manager

Michel Koskas[*]

Résumé

Cet article décrit un nouvel algorithme[1] permettant de gérer des bases de données. Son champ d'application le plus naturel est néanmoins le datawarehouse (OLAP). Il repose sur une représentation dénormalisée de la base. Les données sont stockées dans des thesaurus et des arbres à préfixes (une représentation hiérarchique de champs de bits) qui ont des propriétés intéressantes.

Mots-clefs : base de données, champs de bits, arbres à radicaux, stockage hiérarchique

Abstract

This paper describes a new algorithm[2] dealing with databases. This algorithm allow to fully manage a database, but their most natural field of applications is the datawarehouse (OLAP). It lies on a de-normalized representation of the database. The data is stored in thesauruses and radix trees (a hierarchical representation of bitmaps) which have interesting properties.

Key words : database, bitmaps, radix trees, hierarchical storage

1 Introduction

It is often said that database sizes grow by a rate of 10 % a year and that this growth is greater than the one of the abilities of computers. Databases are more and more used

[*] LAMSADE, Université Paris-Dauphine, 75775 Paris cedex 16, France. koskas@lamsade.dauphine.fr

[1] brevet en cours de dépôt

[2] patent pending

in more and more fields: social actors, armies, commercial agents use more and more data. A pertinent use of an enormous amount of data may show a huge profit to the data owner. For instance the “wallmart” stores managers realized, thanks to data mining, that on saturdays, customers who bought pampers for babies usually also bought beer. They re-arranged their stores in order to put aside the beer and the pampers. The result was that the sales of both these articles rose up. (The admitted explanation is that on saturday, it is more often men who make home shopping.)

Usually, to deal with databases, one may use multidimensional arrays, special indexes (bitmaps), relation caching, optimized foreign key joins, B-trees or approximation. The algorithm presented in this paper uses radix trees. It shall be denoted the A-algorithm. These trees may be understood as a hierarchization of bimaps vectors. It allows one to answer to SQL queries or to manage the base (to add or remove tuples, a primary key, a foreign key or an attribute from a relation or even to add or remove a relation to or from a database).

We show in the next relation a comparison between programs written in C++ designed to use these algorithm and the same requests performed on the same machine but using the three most popular commercial products allowing databases management. The requests were taken from the TPC (see [6]).

When one deals with databases, one may have to answer to two very different kind of queries: one of them is “What is the content of attribute C in the relation T at the record Id 17?” and the other is “At which record Ids may I find this given tuple for the attribute C in relation T?”.

The first query may be very easily answered by reading the relation the wanted attribute belongs to.

But for this first request, there is a case in which the answer is not that easy. This is the case when the attribute C does not belong to T but to a relation T' linked to T by foreign keys and primary keys.

One may answer this kind of request (“What is the content of tuple with record Id 17 of attribute C in relation T with C not belonging to T”) by using a de-normalized data representation.

One may also answer very easily to a request like “Where may I find this given tuple in attribute C in relation T?” by using radix trees (radix trees may be seen as a hierarchical representation of bitmaps indexes). The bitmaps are widely used in database management. One may refer to [10] for a recent work in this matter.

The data of databases is very often stored in B-trees (see [4], [3] or [5] for instance.).

The complexity of the computation of an "and" request with the A-algorithm is averagely $O(i \ln L)$ where i is the cardinality of the intersection and L the maximum of the cardinalities of the numbers of records of the relations involved in the request. In the worst case (whose probability of appearance tends to 0 when the size of the data tends to infinity), this complexity of this computation is $O(L \ln L)$. This is the complexity of the algorithms using balanced trees for instance.

The complexity of an "or" request with the A-algorithm is $O(L \ln L)$, which is also the case of the use of B-trees. The complexity of insertion, suppressions or updates are, with the A-algorithm, $O(\ln L)$. These operations are also performed in $O(\ln L)$ with algorithms using B-trees.

The reader may refer to [8] or [9].

1.1 Plan of the paper

The paper is organized as follow: the section 1 introduces the problem discussed in this paper. Its plan is the present subsection (1.1). The next subsection is dedicated to a presentation of the TPC benchmark (1.2) and the performances of the A-algorithm are presented in 1.3.

The next section is devoted to an introduction to radix trees (2). The two next subsections (2.1 and 2.2) explain how one can perform set operations over radix trees.

The next section deals with the creation of the indexes of a database using radix trees (3). A fundamental case is when the database is made of a single relation itself containing a single attribute. The subsection 3.1 explains it. The next subsections detail the creation of the thesaurus (3.1.1), the storage of the indicative functions (the sets of records ids of each word of the thesaurus are stored in radix trees, subsection 3.1.2). It is convenient to use macro words to accelerate the computations of between clauses (subsection 3.1.3). The two next subsections give details of the storage of the attribute (3.1.4 and 3.1.5) and the next one summarizes the storage of an attribute (3.1.6).

The two subsections are dedicated to cases when a relation has several attributes (3.2) of when the database has several relations (3.3).

Once the indexes are built, one may request the database (section 4). The first step is to compute the expansion relation and to remove the join clauses (4.1). The atomic requests are treated in the subsection 4.2. An important case is the between (4.2.1) because it has several sub cases (4.2.2, 4.2.3, 4.2.4, 4.2.5). Then one may mix these atomic cases to perform any “where” clause which does not contain sub requests (4.3). Its logical connectors are the “or” (4.3.1), the “and” (4.3.2) and the “not” (4.3.3). A more problematic case is the case of comparison between attributes (4.4) which is very similar to a cartesian product (4.5) Then one has to manage the sub queries when they are correlated (4.6) or not obviously (4.7). The last step is to perform computations on the tuples which are at the record ids found in the “where” clause (4.8).

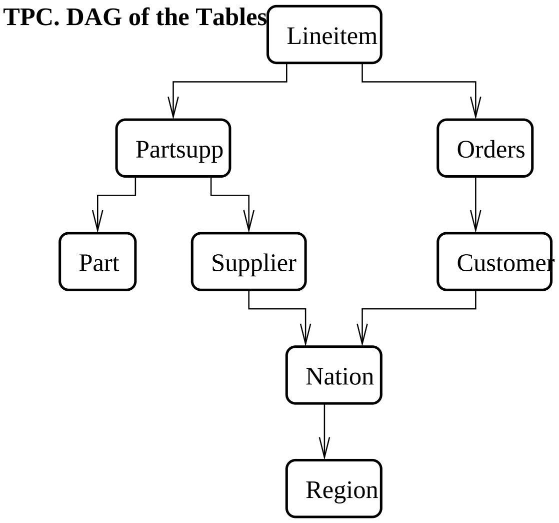


Figure 1: The dag of the relations of the TPC

The next section deals with the base management (5). One may manage a relation (5.1) by adding or removing records (5.1.1 and 5.1.2), add or remove an attribute to a relation (5.1.3 and 5.1.4), add or remove a primary key or a foreign key (5.1.5, 5.1.6, 5.1.7, and 5.1.8), add or remove a relation (5.2 and 5.2.1)

The before to last section (6) is the conclusion and the last section is dedicated to aknoledgements (7).

1.2 The TPC

The TPC (the Transaction Processing Performance Council, see [6]) is a benchmark designed to measure the performances of database manager programs. One can download a relational database made of eight relations: Lineitems, Partsupp, Part, Supplier, Orders, Customer, Nation and Region. This base may be scaled by a scale factor as big as 1000. When its tuple is 1, the size of the database is roughly 1 GB. In this case, the relation Lineitem is made of 6 millions lines, Partsupp of 800,000 lines, Part of 200,000 lines, Supplier of 10,000 lines, Orders of 1,500,000 lines, Customer of 150,000 lines, Nation of 25 lines and Region of 5 lines. The dag of the relations is as follow (an arrow between two relations T_1 and T_2 from T_1 to T_2 means that the relation T_1 contains a foreign key replicating a primary key of T_2 (see figure 2).

The attributes of the relations were the following:

The tpc benchmark contains 22 queries : 20 of them are queries of the data and the two last queries are insertion and suppression of 10% of the lines of `lineitem`.

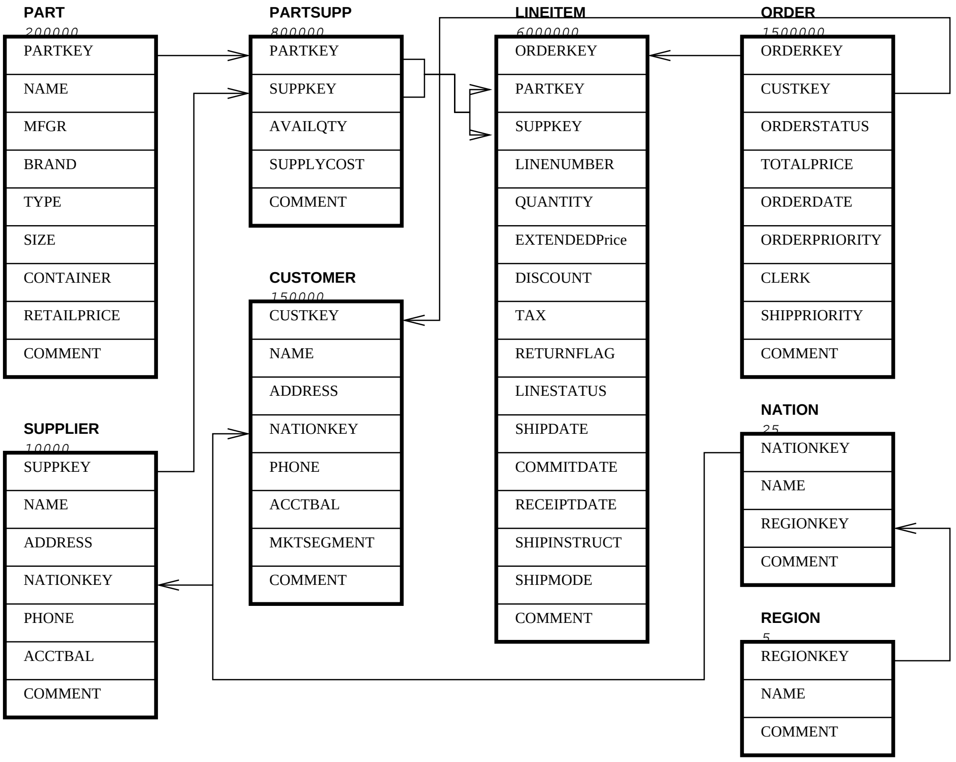


Figure 2: The attributes of the relations of the TPC

The queries performed for this paper where the $Q1$, $Q6$, $Q17$ and $Q19$. These requests are:

Q1:

```
select
    l_returnflag,
    l_linestatus,
    sum(l_quantity) as sum_qty,
    sum(l_extendedprice) as sum_base_price,
    sum(l_extendedprice * (1 - l_discount)) as
sum_disc_price,
    sum(l_extendedprice * (1 - l_discount) * (1 +
l_tax)) as sum_charge,
    avg(l_quantity) as avg_qty,
    avg(l_extendedprice) as avg_price,
    avg(l_discount) as avg_disc,
    count(*) as count_order
from
    lineitem
where
    l_shipdate <= date ’1998-12-01’ - interval ’:1’ day
(3)
group by
    l_returnflag,
    l_linestatus
order by
    l_returnflag,
    l_linestatus;
```

Q6:

```
select
    sum(l_extendedprice * l_discount) as revenue
from
```

```
    lineitem
where
    l_shipdate >= date ’:1’
    and l_shipdate < date ’:1’ + interval ’1’ year
    and l_discount between :2 - 0.01 and :2 + 0.01
    and l_quantity < :3;
```

Q17:

```
select
    sum(l_extendedprice) / 7.0 as avg_yearly
from
    lineitem,
    part
where
    p_partkey = l_partkey
    and p_brand = ’:1’
    and p_container = ’:2’
    and l_quantity < (
        select
            0.2 * avg(l_quantity)
        from
            lineitem
        where
            l_partkey = p_partkey
    );
```

Q19:

```
select
    sum(l_extendedprice* (1 - l_discount)) as revenue
from
    lineitem,
    part
```

```
  where
      (
          p_partkey = l_partkey
          and p_brand = ’:1’
          and p_container in (’SM CASE’, ’SM BOX’, ’SM PACK’, ’SM PKG’)
          and l_quantity >= :4 and l_quantity <= :4 + 10
          and p_size between 1 and 5
          and l_shipmode in (’AIR’, ’AIR REG’)
          and l_shipinstruct = ’DELIVER IN PERSON’
      )
      or
      (
          p_partkey = l_partkey
          and p_brand = ’:2’
          and p_container in (’MED BAG’, ’MED BOX’, ’MED PKG’, ’MED PACK’)
          and l_quantity >= :5 and l_quantity <= :5 + 10
          and p_size between 1 and 10
          and l_shipmode in (’AIR’, ’AIR REG’)
          and l_shipinstruct = ’DELIVER IN PERSON’
      )
      or
      (
          p_partkey = l_partkey
          and p_brand = ’:3’
          and p_container in (’LG CASE’, ’LG BOX’, ’LG PACK’, ’LG PKG’)
          and l_quantity >= :6 and l_quantity <= :6 + 10
          and p_size between 1 and 15
```

Request	A-algo	DBM1	DBM2	DBM3
Q1	8s	47s	370s	33s
Q6	2s	24s	22s	24s
Q17	3s	8s	10s	8s
Q19	3s	19s	24s	25s
RF1 (Insert)	4s	231s	96s	53s
RF2 (Delete)	5s	121s	85s	42s
Cartesian Product	7s	non Op.	Non Op.	Non Op.

Table 1: Performances of the A-algorithm compared to SQL Server 2000, Oracle 8i and DB2

```
          and l_shipmode in ('AIR', 'AIR REG')
          and l_shipinstruct = 'DELIVER IN PERSON'
      );
  p_partkey = l_partkey
          and p_brand = ':3'
          and p_container in ('LG CASE', 'LG BOX', 'LG
PACK', 'LG PKG')
          and l_quantity >= :6 and l_quantity <= :6 + 10
          and p_size between 1 and 15
          and l_shipmode in ('AIR', 'AIR REG')
          and l_shipinstruct = 'DELIVER IN PERSON'
      );
```

1.3 Performances

The comparison between programs using the algorithm detailed in this paper and the main programs one can buy were all performed on the same PC, using a single processor of 2GH, 1GB of RAM, and all the programs written in C++ ; the data was the TPC data using a scale factor of 1, so the size of the database (the flat relations) was roughly 1 GB. The cartesian product was performed over two copies of the main relation of the TPC, the relation lineitem, holding 6 millions lines. DBM1 is Microsoft SQL Server 2000, DBM2 is Oracle 8 and DBM3 is IBM DB2).

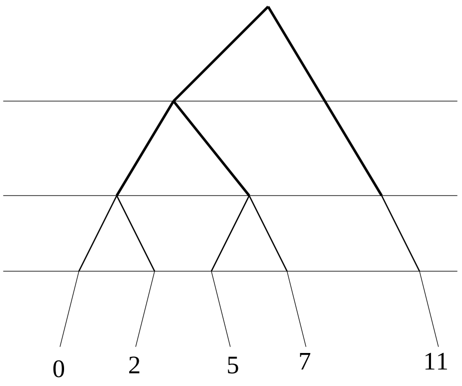


Figure 3: The set $\{0, 2, 5, 7, 11\}$ stored in a radix tree

The algorithm is based on a full use of hierarchical data representation (by the use of radix trees), for the data and the record Ids they belong to.

In a first part we recall the use of radix trees. This tool will be very useful to fully manage the database.

2 Radix trees

A radix tree is a convenient mean to store a set of integers or words of a dictionary, especially when they have all the same length. When dealing with integers, one can manage to force them to have the same length, by adding prefixes made of repeated 0s. A radix tree stores its elements in its leaves. When storing numbers written in basis 2, the nodes may only have a left son (labeled with 0) or a right son (labeled with 1). The path between the root of the tree and one of its leaves writes a word onto the alphabet $\{0, 1\}$ and this word form the digits of the stored integer.

Let us then consider for instance a set of integers written in basis 2 and of same length in this basis, S. One can store S in a tree whose paths between the root and the leaves are the integers of S. For instance, the set $S = \{0, 2, 5, 7, 11\} = \{0000, 0010, 0101, 0111, 1011\}$ may be stored as (see figure 3).

The advantages of storing a set of integers in such a way are numerous: the storage is efficient because common prefixes are stored only once in the tree, and, as we will see in the next subsections, the computations over sets of integers are quite easy to perform and efficient.

An algorithm to build a radix tree whose leaves are the elements of a set S is the following:

Algorithme 1

```
RadixTree Build(set S, height H)
Parameters: a set S and $H$ = 1 + $\lceil \ln_2(Max(S)) \rceil$[3]
Result: a Radix Tree
Result = Node
if (H = 0) return Result.
Build $S_0 = S \cap [0, 2^{H-2} - 1]$ and $S_1 = S \cap [2^{H-2}, 2^{H-1} - 1]$
if ($S_0 \neq \emptyset$) Result->LeftNode = Build($S_0$, H-1)
if ($S_1 \neq \emptyset$) Result->RightNode = Build($S_1 - 2^{H-2}$, H-1)
return Result
```

2.1 Intersection

Let S and S' be two sets of integers. We wish to compute the intersection $S \cap S'$ and let us denote s and s' their cardinalities.

One way to do it is to sort the two sets (which costs $O(s \ln s + s' \ln s')$and to compute this intersection of the sorted sets in time $O(max(s, s'))$. So this intersection may be computed in a time like $O(s \ln s + s' \ln s')$.

One may also sort only one of the sets, say S and look for every element of S' in the sorted set S. The cost is like $O(s \ln s + s' \ln s) = O((s + s') \ln s) \leq O(s \ln s + s' \ln s')$.

Now if we suppose that S and S' were stored in radix trees, the cost of the intersection is like $O(i \ln s)$ where $i = \#(S \cap S')$, where $\#(S)$ is the cardinality of the set S. Indeed, the intersection between the two radix trees may be performed level by level

2.2 Union

The cost of computing the union of two sets of integers, S and S', of cardinals s and s' is the cost of making a multi-set union plus the cost of computing the intersection $S \cap S'$ in order to remove the common elements to S and S'.

[3] the parameter H has got not to be re-computed at each recursive call

Here again, one can begin by sorting the two sets and then compute $S \cup S'$. The cost of this algorithm is $O(s \ln s + s' \ln s' + s + s') = O(s \ln s + s' \ln s')$.

In a similar manner, one may sort one of the sets, say S and compute $S \cup S'$ by looking for each element of S' in S. The cost of this algorithm is $O(s \ln s + s' \ln s) = O((s + s') \ln s)$.

Now if we suppose again that S and S' are stored in radix trees, the cost of the computation of $S \cup S'$ is $u \ln s$ where u is the cardinal of $S \cup S'$. Indeed, the two trees may be read simultaneously and the resulting tree may be computed on the flight.

3 Creating indexes

In this section we will explain how one can use radix trees to build convenient indexes to store and manage databases.

In a first subsection, we will suppose that the database is made of only one relation which contains only one attribute. This case, though artificial, is fundamental to understand the proceed described in this paper.

Then we will suppose that the database is composed of one single relation, made of several attributes and at least one Primary Keys. It may be convenient to suppose that a relation may contain several Primary Keys. Indeed, in practice, it may so happen that a Primary Key, made of several attributes, could be only partially filled while another could be fully filled.

The last subsection we be dedicated to the indexes creation of a full database.

3.1 One relation, one attribute

Primary Keys. A primary key is an attribute, or a set of attributes such that two different tuples of the relation may not have the same tuples on this attribute (or all these attributes).

There is one implicit and convenient Primary Key in any relation : the record Id (it is indeed a Primary Key because no two different lines have the same record Id). So we will assume that the tuples of the relation are identified by their record Ids.

If one has to store, request and manage a date base made of one single relation made of only one attribute, one may compute the thesaurus of the attribute and then, for each word of this thesaurus compute the set on integers it appears at.

Then each set may be stored in a radix tree as explained above.

0	Male
1	Female
2	Female
3	Male
4	Female
5	Male
6	Male
7	Female
8	Female
9	Male
10	Male

Table 2: An example of simple relation

3.1.1 Thesaurus creation

Let us notice that this step necessitates a sort : one has to build the set of couples (word, record Id), which is sorted according to the first element and according to the second for the couples which have the same first element. Then one builds on the thesaurus and the set of record Ids each of these words appear at.

Let us take an example: let us consider the following relation (see table 2).

(In this example, the record Ids are indicated explicitly.)

One builds the couples (Male, 0), (Female, 1), (Female, 2), (Male, 3), (Female, 4), (Male, 5), (Male, 6), (Female, 7), (Female, 8), (Male, 9), (Male, 10)

and sorts them according to the first element of the couples:

(Female, 1), (Female, 2), (Female, 4), (Female, 7), (Female, 8), (Male, 0), (Male, 3), (Male, 5), (Male, 6), (Male, 9), (Male, 10).

Then one is able to build the thesaurus and, for each word of the thesaurus, the set of record Ids this word appears at:

“Female” appears at record Ids $\{1, 2, 4, 7, 8\}$ and “Male” appears at record Ids $\{0, 3, 5, 6, 9, 10\}$.

When this is done, it is easy to answer a request like "What are the record Ids the word “Male” appears at?", but quite uneasy to answer to the request “what is the tuple at record Id 7?”. For this last request, see subsection 5 below.

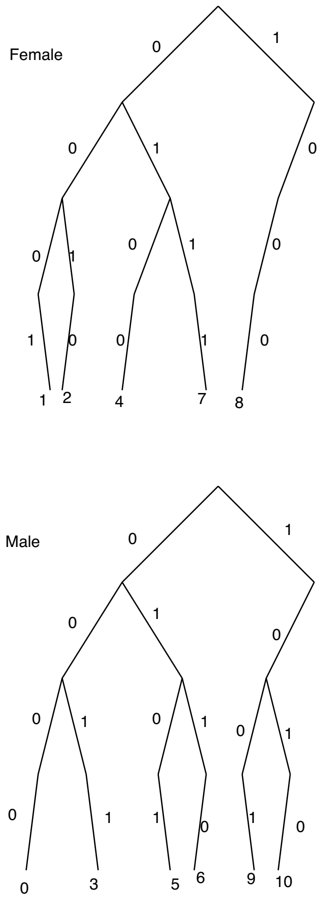


Figure 4: Example of the representation of an attribute of a relation

3.1.2 Storing the indicative functions

Now these sets of record Ids each word of the thesaurus appear at can be stored in radix trees. This is convenient and powerful to compute intersections, and so on…

In the preceding example, one has: (see Figure 4)

3.1.3 Creating macro-words

Another question one may have to answer to when dealing with the attribute of a relation of a database is a between: one may want to know for instance for which record Ids the words lye between two given values.

Let us imagine for instance that an attribute is made of dates, formated in YYYYMMDD. Compare two dates is compare lexicographically the two words.

But we may also enlarge the thesaurus, with words that are truncates of the initial words. Let us indeed add words to the thesaurus of the attribute, for instance any truncate of six characters or any truncate of four characters.

Then any word of the thesaurus will be represented three times: one time as itself, one time as a truncate of six characters and one time as a truncate of four characters.

Any word of six characters, say aaaamm, will occur each time a word aaaammxx occurs. In other words, the record Ids the word aaaamm appears is exactly the union of the sets of record Ids any word aaaammxx appears at.

In a similar manner, any word of four characters, say aaaa, will occur each time a word aaaaxxyy occurs and its radix tree will be the union of the corresponding radix trees.

In summary, one builds not only the radix trees of each word of the thesaurus but also the thesaurus of each prefix of given lengths of words. So when one has to solve a "between" clause, one splits the wanted interval with respect to the prefix length pre-computed and read the matching radix trees. For instance, the interval $[19931117, 19950225]$ would demand, without the macro words, 466 reading of radix trees because this interval contains 466 different words. If one splits this interval with respect to prefix lengths of 6 and 4, one has: $[19931117, 19950225] = [19931117, 19931130] \cup [199312, 199312] \cup [1994, 1994] \cup [199501, 199501] \cup [10050201, 19950225]$. The first interval contains 14 different words (not truncated). The second contains a single truncated word (6 characters), and the reading a single radix tree gives the set of the records Ids words like $199312dd$ appear at. The third interval contains one single truncated word (4 caracters) and the reading of the single matching radix tree gives the set of records Ids words like $1994mmdd$ appear at, and so on... Finally only 42 readings of radix trees are made necessary instead of 466.

3.1.4 Managing lacks

Now, it may also so happen that some tuples were not filled. But each must have an attribute, even an attribute meaning that there is no attribute at this record Id.

The tuples meaning a lack of information should be chosen in a way as few disturbing as possible, which means we should choose very seldom tuples. We may for instance chose : $\#Empty\#$ for a string, -2^{31} for a signed integer on 32 bits, $2^{32} - 1$ for an unsigned integer on 32 bits, $-2^{6}3$ for a signed integer on 64 bits, $2^{64} - 1$ for an unsigned integer on 64 bits and so on...

0	Male
1	Female
2	Female
3	Male
4	Female
5	Male
6	Male
7	Female
8	Female
9	Male
10	Male

Table 3: An attribute

0	Female
1	Male

Table 4: The thesaurus

3.1.5 An additional storage

As explained above, the storage of an attribute by thesaurus and radix trees makes quite uneasy to answer a question like “what is the tuple at record Id 17?” for instance.

This is why it is necessary to store the attribute in its natural order. Of course, instead on storing the attribute itself, it may be much more affordable to store the word indexes in the thesaurus.

For instance, the preceding attribute shall be stored:

shall be stored as

and the attribute:

remark it sometimes happen that a word could appear or disappear from a thesaurus while adding or removing records to a relation. In this case, we might think we have to rewrite the whole attribute each time this situation happens. This is nevertheless not true: one may store an unsorted thesaurus and a permutation which stores is contents. Thus when words are no longer in the s-thesaurus or when a new word appears in it, on has only to re-write the permutation instead of the whole attribute.

0	1
1	0
2	0
3	1
4	0
5	1
6	1
7	0
8	0
9	1
10	1

Table 5: An cheap storage of the attribute

3.1.6 Summary of the full storage of an attribute

3.2 One relation, several attributes

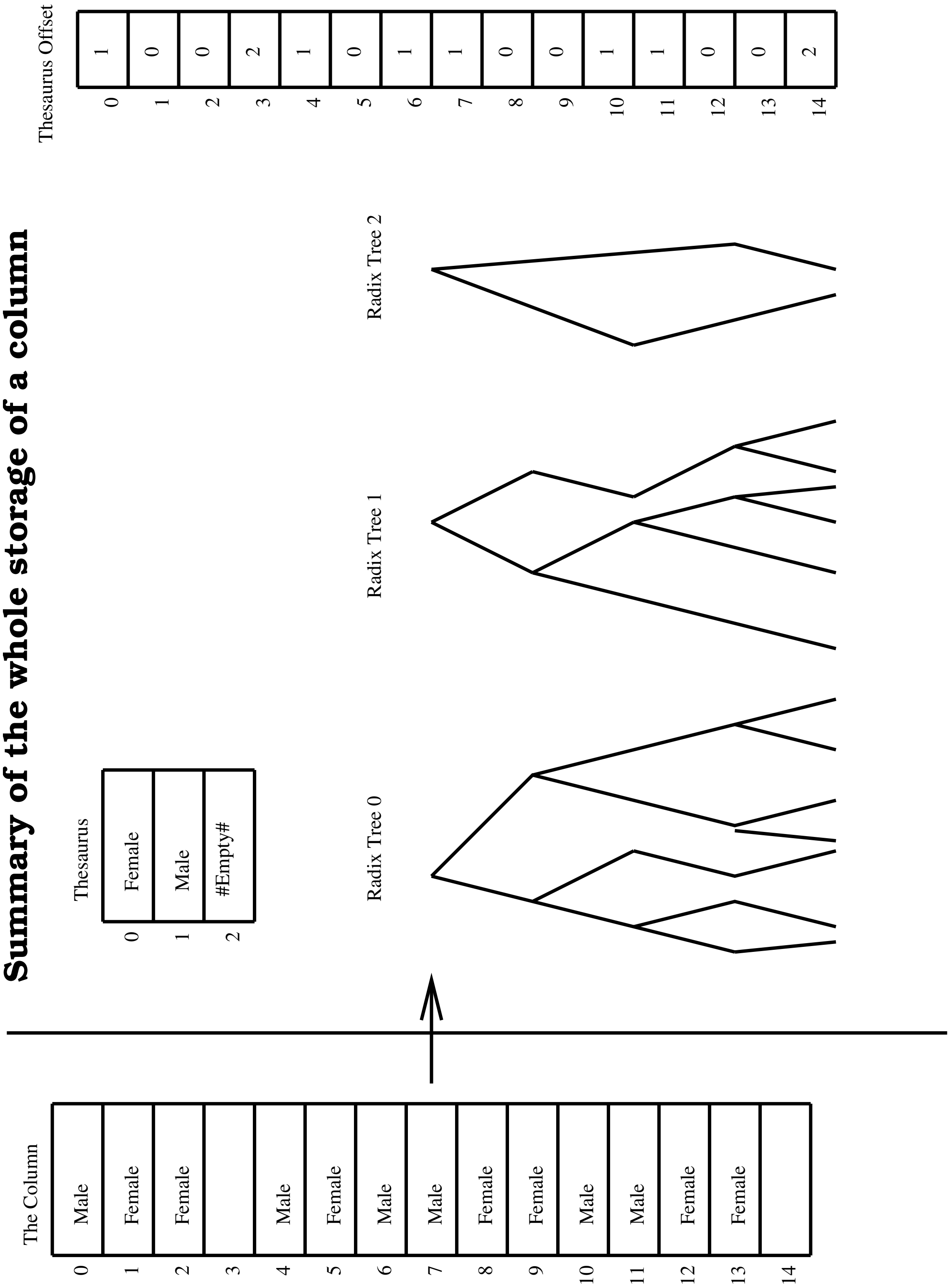
Now when a relation has several attributes, each one of them may be treated as if it were the only attribute of the relation. *This means to say that there should exist a thesaurus for each attribute and the matching radix trees for all word of any of these thesauruses.*

The only remaining question is the storage of the primary keys.

When dealing with a Primary key, one has to be able to answer efficiently to two questions: at what record Id can we found a given tuple of a primary key, and what is the tuple of the primary key at a given record Id.

One may answer efficiently to both these questions by storing the attribute or the attributes of the primary key in its (or their) natural order, namely with increasing record Ids and by storing in more a permutation allowing one to find a given tuple efficiently.

For instance, let us imagine a primary key made of two attributes, whose tuples are:



The Column

0	Male
1	Female
2	Female
3	
4	Male
5	Female
6	Male
7	Male
8	Female
9	Female
10	Male
11	Male
12	Female
13	Female
14	

Thesaurus

0	Female
1	Male
2	#Empty#

Thesaurus Offset

0	1
1	0
2	0
3	2
4	1
5	0
6	1
7	1
8	0
9	0
10	1
11	1
12	0
13	0
14	2

Figure 5: Summary of the whole storage of an attribute

(0)	1	3
(1)	2	1
(2)	3	2
(3)	2	3
(4)	1	2
(5)	3	7
(6)	2	2
(7)	1	1
(8)	3	3
(9)	4	3

In this example, the record Ids are still explicitly expressed between parentheses. One then store these two attributes exactly as they are and a permutation. To store the permutation, one has to chose a comparison function. For instance one may compare first the first attribute and the second in case of equality.

In this case, the sorted primary key is:

(7)	1	1
(4)	1	2
(0)	1	3
(1)	2	1
(6)	2	2
(3)	2	3
(2)	3	2
(8)	3	3
(5)	3	7
(9)	4	3

By removing the tuples (but keeping the record Ids) one obtains the permutation (7401632859) and thus is able to find a given value by dichotomy.

When storing a whole relation, it is also convenient to store the number of its records.

3.3 Several Relations

In a relational database, there are usually several relations linked between them by foreign keys recalling primary keys.

As we explained, a primary key is an attribute or a set of attributes whose tuple may be an unique identification of the record within the relation (the record Id is a fundamental example of primary key. See [1] or [2]).

Let us suppose that a relation is made of several billion of records, but that some attributes may take only five different tuples (for instance, in a genealogy database, one may want to store for each client the country, the continent the customer was born, the country, the continent where his mother was born and the country and the continent his elder child, if any, was born). Instead of recalling fully the names of all these countries and continents for each record, one may build two other relations, one of countries and another of continents. Then on each record, instead of recalling all these countries and continents, one may recall only the primary key of the relation of the countries for the customer, his mother and elder child if any. And in the relation of the countries, one may also recall only the primary key of the relation of the continents the country belongs to. This storage is much cheaper.

Here is a little example of such a practice:

(li)	cn	Inc	BirCoun	BirCont	MoCoun	MoCont	EldCoun	EldCont
(0)	Dupont	817	France	Europe	Tunisia	Africa	England	Europe
(1)	Gracamoto	1080	Japan	Asia	Japan	Asia	USA	America
(2)	Smith	934	England	Europe	India	Asia	England	Europe
(3)	Helmut	980	Germany	Europe	Germany	Europe	Germany	Europe

(cn means “customer name”, Inc “Income”, “BirCoun “Birth Country”, “BirCont “Birth Continent”, MoCoun “Mother’s birth country”, MoCont “Mother’s birth Continent”, EldCoun “Elder’s birth country” and EldCont “Elder’s birth continent”.)

This relation may be rewritten in several relations:

Continents:

Continent	
(li)	Continent
(0)	Africa
(1)	America
(2)	Asia
(3)	Europe

Country		
(li)	Country	Continent
(0)	France	3
(1)	Tunisia	0
(2)	England	3
(3)	Japan	2
(4)	USA	1
(5)	India	2
(6)	Germany	3

And the customers' relation becomes thus:

Customers					
(li)	cn	Inc	BirCoun	MoCoun	EldCoun
(0)	Boyer	817	0	1	2
(1)	Gracamoto	1080	3	3	4
(2)	Smith	934	2	5	2
(3)	Helmut	980	6	6	6

and the set of three relations is indeed much shorter to store than the full relation.

But this also points out that any relational database may be seen as a set of independent relations.

In the preceding example for instance, we can consider the relation continent by itself, the relation country with the relation continent expanded inside and the relation people with the relation country and continent expanded inside (which is the very first relation, the full one, of this example).

These expansion relations are thus:

Expanded Continents	
(li)	Continent
(0)	Africa
(1)	America
(2)	Asia
(3)	Europe

Expanded	**Countries**	
(li)	Country	Continent
(0)	France	Europe
(1)	Tunisia	Africa
(2)	England	Europe
(3)	Japan	Asia
(4)	USA	America
(5)	India	Asia
(6)	Germany	Europe

Expanded Customers								
(li)	cn	Inc	BirCoun	BirCont	MoCoun	MoCont	EldCoun	EldCont
(0)	Boyer	817	France	Europe	Tunisia	Africa	England	Europe
(1)	Gracamoto	1080	Japan	Asia	Japan	Asia	USA	America
(2)	Smith	934	England	Europe	India	Asia	England	Europe
(3)	Helmut	980	Germany	Europe	Germany	Europe	Germany	Europe

Of course, it may so happen, like in this example, that a relation should be expanded several times in another. This means that the attributes of an expanded relation should be refereed to as the attribute of the expanded relation expanded in the expansion relation via the list of couples (Primary Key Foreign Key) allowing one to move from the expansion relation to the expanded relation.

Now we define an expansion relation as a relation in which as much relations as possible were expanded in. From now on, we will consider only expansion relations and the database will be made, from now on, of independent expansion relations.

For each of these expansion relations, one can build the indexes as explained above.

And now, we are ready to request or manage the database.

4 Requesting

In this section we explain how one may use the indexes created as explained below to perform efficient SQL requests. Usually, a request involves several relations. It may be split in two parts: the first part means to discriminate record Ids and the second part (if any) means to perform computations over the data of the found records.

The first part may contain join clauses (the link between a foreign key and the matching primary key), comparison between an attribute and a constant (with arithmetic connectors as “=”, “$\geq$”, “$>$”, “$\leq$”, “$<$”, between, like, in…), or a comparison between two attributes (for instance like in a cartesian product), theses requests being logically connected by logical connectors like “and”, “or”, “not” ….

The second part may contain arithmetic operations like a sum, a mean, a product, a star operator, ….

4.1 Removing the join clauses: choice of the expansion relation

As explained above, each of the relations, say R, is considered as an "expansion relation" which means that any relation R' linked to R via a foreign key are expanded in R.This means that the attributes of R' are developped in R, the thesauruses of these attributes are stored as the ones of R and the matching radix trees are computed. So the join clauses are irrelevant in such a relation.

But a request involves usually several relations. How should we chose the appropriated expansion relation? The relations involved in the request are all expanded in a nonempty set of relations, say $\mathcal{T}$. Exactly one of these relations is expanded in none of the others. This relation is the expansion relation appropriated to solve the request.

Now, the where clause may contain some join clauses. These clauses must be logically linked to the remaining part of the request by an "and" operator. So the first step consists in simply remove these clauses by replacing the (Join Clause And Remaining) part by (Remaining).

Now let us study how we can manage the where clause cut down from its join clauses

4.2 Atomic requests

We call here an atomic request a fundamental part of a where clause, namely a comparison clause linked to the remaining of the where clause by logical operators. If t is a relation and c one of its attributes, an atomic clause may be `t.c = 3`, `t.c between "HIGH" and "MEDIUM"`, or `t.c like Word%` for instance.

We explain in the above subsections how to deal with the atomic requests.

4.2.1 Equality between an attribute and a constant

This is the simplest case: one has only to find the wanted value in the thesaurus, read its radix tree which gives him the record ids this word appear at.

4.2.2 Between

This is the fundamental example of atomic request. Any of the others may be treated as a between. It is the clause the macro words where made for.

Let us take back the "date" example given below: one made macro words of length 4 and 6 for an attribute containing dates and wishes to compute the record Ids dates lying in $[19931117, 19950225]$ appear at.

As explained above, one may split the interval by same common prefixes length than the macro words lengths. Thus, one obtains $[19931117, 19950225] = [19931117, 19931130] \cup [199312, 199312] \cup [1994, 1994] \cup [199501, 199501] \cup [10050201, 19950225]$.

The computation is then simple: one read the radix tree of the 19931117, "OR" it with the radix tree of $19931118, \ldots$, "OR" the result with the radix tree of 199312 (the macro word whose radix tree is precisely the "OR" of the radix tree of all the dates beginning with 199312), then "OR" the result with the radix tree of the macro word 1994 (whose radix tree is the "OR" of the radix trees of all the dates beginning with 1994 and so on....

As explained above, one reads only 42 radix trees to perform this computation instead of 466...

Of course one can also manage opened or semi opened intervals by simply excluding the corresponding word.

4.2.3 Greater than, lower than, greater than or equal, lower than or equal

Each of these atomic requests is in fact a between. Indeed, if we call `m` the minimum value of the thesaurus and `M` its maximal value, then any of these requests are either of the form (m, a) or of the form (a, M). So if we can manage the between clause, we can also manage these atomic requests.

4.2.4 In

The `in` clause is a way of mixing equality clauses and Or logical connections. So we can manage them simply.

For instance, `t.c in (a, b, c)` may be rewritten in `t.c = a or t.c = b or t.c = c`. The management of the `or` clause is explained below.

4.2.5 Like

The like clause is another example of between clause: for instance, the clause `t.c like word%` may be rewritten in: `t.c between [word, wore[`. Here again, manage the `between` clause also manages the `like` clause.

4.3 Mixing atomic requests

Now the where clause may mix atomic clauses by using logical operators: the `or`, the `and` and the `not` clause. The three next subsections are dedicated to these logical clauses.

We would like to empathize that the result of an atomic request is a radix tree.

We will suppose (and show) that this is the case of any where clause.

4.3.1 Or

Now we have to OR two radix trees: The clause is `(Left Clause OR Right Clause)`. The `Left Clause` and `Right Clause` when solved, return a radix tree. So all we have to do is to compute recursively the resulting radix tree of the full clause.

4.3.2 And

The `And` clause may be performed exactly as the `Or` clause. However, the computation is a little more efficient.

Indeed, we have to `and` two radix trees; this computation is made recursively, checking the matching nodes of the two trees simultaneously. But when one of the trees contains a node and the other tree does not contain the matching node, it is of course irrelevant to perform the and of the sons of this node.

4.3.3 Not

the `not` clause is the most difficult atomic clause to perform with radix trees.

Each relation's size (its number of records) is stored. So perform a `not` over a radix tree may be done as follow: (the goal is to perform `not` T with T a radix tree).

let us define a n-full radix tree (n-frt) as a tree designed to contain all the numbers from 0 to $n-1$.

Then to perform a `not`, one may go from a n-frt (where n is the number of records of the expansion relation the request is solved onto) and remove the nodes corresponding to T.

To remove a node, one may proceed by removing the node and removing recursively its father if it has any child left.

For instance, if the expansion relation has 13 records, the `not T` with `T` the following tree (see Figure 6)

is (see Figure 7)

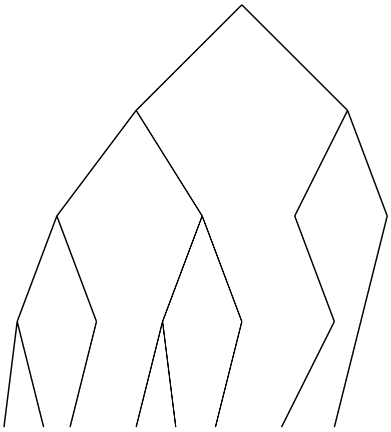


Figure 6: A 13-radix tree before a NOT operation

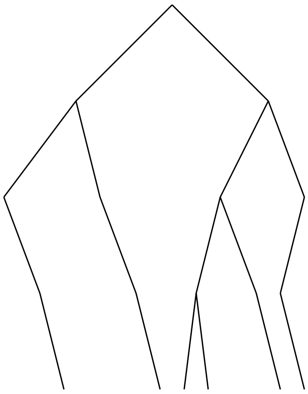


Figure 7: The same 13-radix tree after the NOT operation

4.4 Comparison between attributes

The comparison between two attributes is the most complex request to perform with this data representation and has a lot to do with the cartesian products section just below).

Let `t` be the expansion relation of the request and `c` and `d` be two of its attributes. A comparison between attributes may be a part of a where clause in which we discriminate the records such that for instance `t.c > t.d`. We empathize the fact that this comparison is done at the same record Ids (this is the difference with the cartesian product).

So how can we perform this clause?

Let $\mathcal{T}_c$ and $\mathcal{T}_d$ be the thesauruses of the attributes `t.c` and `t.d`. We are looking for the record Ids such that `t.c > t.d`. Here is how we may proceed. For each word w of the thesaurus $\mathcal{T}_c$, we can compute the radix tree r of the interval $[m_d, w']$ where w' is the greatest word of $\mathcal{T}_d$ lower than w. Then by performing an `and` over the w's radix tree and r, one obtains the corresponding record Ids for w.

By `Or`-ing the results of all the words of $\mathcal{T}_c$, one obtains the wanted radix tree.

Let us notice that the radix trees r (as above) are not to be computed independently one of the others: if the words w are read in an increasing order, one simply has to `Or` the radix trees corresponding to the intervals $[w_i, w_{i+1}[$.

The other such clauses may be performed in a similar way.

4.5 Cartesian products

The cartesian product is usually considered as a combinatoric computation over relation of a relational database. Actually, this computation may be performed quite simply and efficiently. The authors performed for instance a cartesian product of two relations (both of them of 6 millions records) in 7 seconds on an average computer.

Let us consider such a request: compute the number of times `t.c > t'.d` independently of the record Ids.

For instance, if the attributes `t.c` and `t'.d` are:

`t.c`
`z`
`ab`
`z`
`c`
`da`
`e`

and

`t.d`
`b`
`a`
`as`
`sa`
`ca`
`ba`
`abra`

then the number of times `t.c > t.d` is $7+1+7+5+6+6=32$. The complexity of the naive algorithm is a constant times the product of the two relations' size. It is not possible to actually perform cartesian products of two relations with modern machines in using this algorithm.

So how can we compute efficiently this result?

The attributes are stored with their thesaurus and the radix trees corresponding to each of the words of these thesauruses.

We may, to each word of the thesaurus, compute the number of records it appears at by a simple reading of the radix trees.

In the preceding example, this gives:

t.c	
ab	1
c	1
da	1
e	1
z	2

t.d	
a	1
abra	1
as	1
b	1
ba	1
ca	1
sa	1

One can also, for the attribute `t'.d` compute, for each word the number of words lower than or equal to it. This gives:

t.d, cumulated cardinalities	
a	1
abra	2
as	3
b	4
ba	5
ca	6
sa	7

Then, by reading the thesauruses and the corresponding numbers, one can compute the result. For each word `w` of `t.c`, one has to look for the greatest word `w′` of the thesaurus of `t′.d` lower than `w` and add to the result the product of the number of occurrences of `w` multiplied by the cumulated number of occurrences of `w′`.

This gives:

w	w′	w-card	w′-cumul. card.	product	partial result
ab	a	1	1	1	1
c	ba	1	5	5	6
d	ca	1	6	6	12
e	ca	1	6	6	18
z	sa	2	7	14	32

This algorithm's complexity is a constant time the sum of the sizes of the thesauruses, which is usually much less than the product of the relation's number of records even if the relations do not contain twice any word (a sum of thesauruses sizes is to be compared to the product of relations sizes. . .)

4.6 Correlated sub queries

This subsection and the following are dedicated to sub-queries. Indeed, the where clause may contain other where clauses and these sub queries may be or not correlated to the principal one.

What is a correlated sub query? An example of such request is the request 17 of the TPC. This request is:

```
select
    sum(l_extendedprice) / 7.0 as avg_yearly
from
    lineitem
    part
where
    p_partkey = l_partkey
    and p_brand = ’[BRAND]’
    and p_container = ’[CONTAINER]’
    and l_quantity < (
        select
            0.2 * avg(l_quantity)
        from
            lineitem
        where
            p_partkey = p_partkey
    );
```

In this request, one has to perform the computation of the sub query in taking into account the condition requested in the principal part of the query (because the `p_partkey` of the sub-query belongs to the principal part of the request).

So this kind of requests may be rewritten in order to have to perform a non correlated sub query. The preceding query would thus become:

```
select
    sum(l_extendedprice) / 7.0 as ag_yearly
from
    lineitem
    part
where
    p_partkey = l_partkey
    and p_brand = ’[BRAND]’
    and p_container = ’[CONTAINER]’
    and l_quantity < (
        select
            0.2 * avg(l_quantity)
        from
            lineitem
            partsupp
        where
```

```
            p_partkey = p_partkey
            and p_brand = '[BRAND]'
            and p_container = '[CONTAINER]'
    );
```

So a correlated sub query may be rewritten in a not correlated sub query. This is the subject of the next sub section.

4.7 General sub queries

Now a SQL request containing not correlated sub queries may be treated simply: each sub query not containing any sub query is treated as a request by itself and the result of the computation takes the pace of the full sub request.

4.8 Perform computations on the found records

Now when dealing with a database, we are able to perform computations of record Ids of the expansion relation (matching the request) according to the `where` clause.

Now let us suppose that the goal of the request is to perform computations over some attributes of the relation but only for the found record Ids. For instance, it may be to compute an average tuple like in the preceding example.

The tuples of any attribute of an expansion relation are stored in the order they appear in it. So it is easy to read this file only for the record Ids matching the first part of the SQL request and perform the requested computation.

5 Managing the database

Now we are able to store a whole database and to perform efficiently SQL requests onto it. Usually, the quickest the SQL requests are performed, the slowest the database is managed.

This is not true in this case: not only are the requests performed fast but the management of the database are also fast (see the table of performances 1 in the introduction of this paper).

Why is that so? The indexes do not contain sorted data, expect the permutations linked to the thesauruses. In particular, there is no stored sorted data upon one or more attributes.

To manage a database, one may wish to add or remove records of a relation of a base, add or remove a primary key, add or remove a foreign key, add or remove a relation. We will see successively these items in the next sub sections.

5.1 Managing a relation

Manage a relation is the most common operation of the management of a database. Indeed, the most usual case is when the database manager wishes to add or remove records of a relation. All the other transformations change the database scheme or organization and these transformations are much more seldom.

Usually, when removing records of a relation we will refuse to reschedule the whole relation. This means that some record Ids will be declared free and the corresponding data will be removed from the expansion relations (we will see how below). So a relation will usually have “holes’ ’ : some record Ids will not be considered as filled by anything. These record Ids shall be stored in a file, containing firstly these record Ids and lastly the first index from which all the record Ids are free.

5.1.1 Adding records to a relation

So we wish to add records to a relation. Let us keep in mind that for us, all the relations of the database are expansion relations.

By reading the files of the free record Ids of this relation, we may assign a record Id to each of these records.

So we complete the records by filling the attributes of the relations that may be expanded in this relation (for instance if an attribute is a foreign key, we read the corresponding data in the corresponding relation).

Then we compute the thesaurus and the radix trees of the records to add to the relation and perform “Or” to the thesauruses and the radix trees of each attribute of the relation.

5.1.2 Removing records to a relation

We have here number of problems to solve: all the relations of our database are expansion relations and we do not want to reschedule the whole relation after to have removed some records of it. (If we did so, we would have to rewrite all the thesauruses, all the radix trees of this relation and of all the relations the first one may be expanded in.)

So we have the Ids of the records of the relation `T` to be removed (these record Ids may have found thanks to a `where` clause). So for each attribute `C` of the expansion relation, and for all the words w of the thesaurus of these records, we build the radix trees that we x-or with the radix tree of w of c.

and then we just store the resulting radix tree in place of the preceding one.

We assume here that we do not have to change anything to the expansion relations in which T was expanded. Indeed, if we remove a record expanded in another relation, we should in this case throw an exception because the delete instruction was illegal.

5.1.3 Adding an attribute

Add an attribute c to an expansion relation consists in several operations. We have indeed to treat the relation T the attribute to be added belongs to and the relations in which T is expanded.

The treatment of T consists in building the thesaurus of c, the radix trees of each word of it and to store this whole stuff.

The treatment of each relation T′ in which T is expanded consists in reading the record Ids of T′ that must be added to T. Then one computes the thesaurus, the radix tree of each word of it and store the whole thing.

Let us take an example.

Related relations

T0		
(li)	c1	fk1
(0)	a	2
(1)	b	1
(2)	c	0
(3)	b	1
(4)	e	2

T1			
(li)	pk1	c2	fk2
(0)	0	S	0
(1)	1	T	1
(2)	2	V	0

T2		
(li)	pk2	c3
(0)	0	X
(1)	1	Y

The corresponding Expansion relations are thus:

Expanded T0							
(T0)			(T1)			(T2)	
(li)	c1	fk1	pk1	c2	fk2	pk2	c3
(0)	a	2	2	V	0	0	X
(1)	b	1	1	T	1	1	Y
(2)	c	0	0	S	0	0	X
(3)	b	1	1	T	1	1	Y
(4)	e	2	2	V	0	0	X

Expanded T1					
(T1)			(T2)		
(li)	pk1	c2	fk2	pk2	c3
(0)	0	S	0	0	X
(1)	1	T	1	1	Y
(2)	2	V	0	0	X

Expanded T2		
(li)	pk2	c3
(0)	0	X
(1)	1	Y

and let us suppose we wish to add an attribute c2 to T2, whose tuples are Y and Z.

The (expanded) relation T2 becomes

Expanded T2			
(li)	pk2	c3	c2
(0)	0	X	Y
(1)	1	Y	Z

To compute the new expanded relation T1, one reads the tuples of the primary key pk2 and copy the matching tuples of T2 in the new attribute of T1. This gives:

T1						
(T1)				(T2)		
(li)	pk1	c2	fk2	pk2	c3	c2
(0)	0	S	0	0	X	Y
(1)	1	T	1	1	Y	Z
(2)	2	V	0	0	X	Y

in a similar manner, one reads the tuples of pk2 in t0 to compute the new expanded relation T0. This gives:

T0								
(T0)			(T1)			(T2)		
(li)	c1	fk1	pk1	c2	fk2	pk2	c3	c2
(0)	a	2	2	V	0	0	X	Y
(1)	b	1	1	T	1	1	Y	Z
(2)	c	0	0	S	0	0	X	Y
(3)	b	1	1	T	1	1	Y	Z
(4)	e	2	2	V	0	0	X	Y

5.1.4 Removing an attribute

Remove an attribute is a simple operation. It only consists in erasing the file corresponding to this attribute for the relation `T` it belongs to and in all the relations `T′` in which `T` is expanded.

5.1.5 Adding a Primary Key

A primary key is stored by the data of the involved attributes and a permutation storing the order of the data of the primary key.

Add a primary key is thus simply to store the data of the involved attributes and the corresponding permutation.

5.1.6 Adding a Foreign Key

Adding a foreign key is quite more complicated.

A foreign key fk, belonging to an expansion relation T is hooked to a primary key pk, belonging to an expansion relation T'. The relation T' must be expanded into T according to the couple (foreign key, primary key) being treated even if this relation is already expanded into T by the mean of another foreign key.

For each record of T, of index i, the foreign key has a tuple v and we can find the record Id $p(i)$ of T' where $pk = v$.

Then we add al the attributes of T' in T. To perform this, for each attribute c of T' we read the tuple $T'.c[p[i]]$ for all integers i. Then we do as usually by building the thesaurus of this attribute and for each word of this thesaurus the matching radix tree.

5.1.7 Removing a Primary Key

Remove a primary key consists in deleting the corresponding files. Of course, if this primary key is the target of a foreign key, we should throw an exception because such an instruction should be illegal.

5.1.8 Removing a Foreign Key

Let us denote fk the foreign key to be removed and T the relation it belongs to. This foreign key targets a primary key, pk, belonging to a relation T'.

To remove a foreign key breaks the link between two relations. This means that T' is no longer expanded in T and in none of the expansions of T.

5.2 Managing the base

Manage the database itself consists in adding or removing a whole relation…

5.2.1 Adding or removing a relation

Adding (removing) a relation consists in adding (removing) all its attributes, all its primary keys and and all its foreign keys. All these algorithms have been explained above.

6 Conclusion

One of the problems of the use of bitmaps is the size of the involved vectors and the fact that usually many of the bits are equal to 0. In this paper we exposed a database manager algorithm. It may be used to fully manage a database with performances showed in the table 1. The use of radix trees seems to be an interesting hierarchization of bitmaps. They show the advantages to make possible an affordable storage of bitmaps, only the parts with 1s are stored. They also allow a computation level by level, which gives good performances in particular ti solve "and" requests.

This algorithm is also parallelizable and a possible future work is to implement a parallel version of the A-algorithm (this work is in progress, in cooperation with Christophe Cérin). This is to be compared to performances obtained with parallel B-trees like in [7], for instance

In order to keep the efficiency of the A-algorithm, one has to pre-compute the join clauses. The use of macro-words makes also faster the resolution of “between” clauses.

The author would like to apply these ideas to related problems, like find all the occurrences of a pattern in an image whatever the foreground would be, or to find a sound in some sounds whatever the noise would be.

7 Acknowledgment

The author would like to thank warmly Dominique Bernardi for numerous and instructive discussions, Christophe Cérin for numerous comments on an earlier version of this paper and give a special thanks to Maude Manouvrier for very interesting and instructive conversations and comments on an earlier version of this paper.

References

[1] Raghu Ramakrishnan and Johannes Geheke, Database Management Systems

[2] Abraham Silberschatz, Henry F. Korth, S. Sudarshan, *Database System Concepts*, McGraw-Hill ISBN 0-07-228363-7.

[3] R. Bayer and E. McCreight, *Organization of Large Ordered Indexes,* Acta Iform. 1 (1972), 173 189

[4] Gaston H. Gonnet, Ricardo Baeza-Yates.*Handbook of Algorithms and Data Structures,*`http://www.dcc.uchile.cl/˜rbaeza/handbook/`

[5] Chris Jernaine, Anindya Datta, Edward Omiecinski. (1999) *A Novel Index Supporting High Volume Data Warahouse Inbsertion.* VLDB Conf. 235-246.

[6] `http://www.tpc.org`

[7] Shogo Ogura, Takao Mura, *Paging B-trees for Distributed Environments*

[8] D. Lomet, *B-tree Page Size When Caching is Considered*, SIGMOD Record (ACM Special Interest Group on Management of Data), 27 (1998), 3, p 28

[9] O'Neill, P. and O'Neil E. *Databases: principles, programming and performance*. 2nd ed., Morgan Kaufman Publishers, San Francisco, CA., 2000.

[10] Tadeusz Morzy, Maciez Zakrzewicz, *Group Bitmap Index: A Structure For Rules Associations Retrieval,* American Association For Artificial Intelligence, 1998.