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

# ANNALES DU LAMSADE N°7
# Mai 2007

## Robustness in OR-DA

Guest Editors**: Bernard Roy, Mohamed Ali Aloulou, Rim Kalaï**

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



Pour se procurer l'ouvrage, contacter 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.

# Éditorial

Les contributions qui constituent le présent volume ont, pour la plupart d'entre elles, donné lieu à présentations orales et débats au sein d'un groupe de travail. Ce groupe, créé en 2003 par Bernard Roy sous l'égide du LAMSADE, avait pour objet de confronter, de comparer et d'approfondir les travaux de recherche ayant trait à la robustesse en recherche opérationnelle et aide à la décision, lesquels devenaient de plus en plus nombreux et diversifiés. Deux réunions d'une journée entière se sont tenues chaque année. Chacune d'elles a rassemblé des chercheurs et enseignants-chercheurs provenant d'universités multiples, essentiellement françaises et belges. Ce groupe de travail a bénéficié d'un soutien dans le cadre du programme de coopération entre le CNRS et le FNRS/CGRI Belgique.

Tous les articles de ce volume ont été soumis, sous le contrôle des éditeurs, à une procédure d'arbitrage mobilisant, pour chacun, deux arbitres. Dans le présent volume, les articles ont été rangés par ordre alphabétique du premier auteur. Toutefois, pour les présenter brièvement, nous avons adopté ci-après un critère de rangement différent qui nous a conduits à les regrouper en quatre familles.

La première rassemble deux articles qui, chacun à leur manière, visent à mettre en évidence l'intérêt du sujet dont traite le présent volume ainsi que la variété des travaux qui lui sont consacrés dans la littérature. Il s'agit des articles de L. Dias et B. Roy.

L'article de L. Dias débute par une rapide présentation des difficultés les plus courantes auxquelles le chercheur opérationnel se heurte pour attribuer une valeur aux paramètres que comportent les modèles d'aide à la décision (décision individuelle ou de groupe). L'analyse de robustesse a pour objet de répondre à ces difficultés. L'auteur décrit trois usages possibles qui mettent en évidence les bénéfices que l'on peut retirer de « l'outil analyse de robustesse » en les faisant intervenir du début à la fin du processus de décision. VIP analysis et IRIS lui servent d'exemples pour illustrer ses propos. L'auteur met l'accent sur la recherche de conclusions robustes plutôt que sur celle de solutions robustes. Il souligne ainsi le rôle de la modélisation comme processus constructif d'apprentissage.

Après avoir défini le sens qu'il attribue au qualificatif « robuste » (en RO-AD), B. Roy explique pourquoi il préfère parler de « préoccupation de robustesse » plutôt que d'analyse de robustesse. Il explicite et approfondit ensuite les multiples raisons d'être de cette préoccupation. Afin de mieux mettre en évidence le caractère multi facettes de cette préoccupation, il propose de regarder les travaux traitant de robustesse en les situant sur trois versants : les deux premiers, qualifiés respectivement de *standard* (S) et *concret* (C), présentent des traits caractéristiques qui les opposent à bien des égards alors que le troisième, *mixte* (M), est destiné à supporter ceux des travaux qui ne présentent que trop imparfaitement tous les traits caractéristiques requis pour pouvoir être clairement affectés à l'un des deux versants précédents. Dans la section 4, il montre que, sur chacun de ces trois versants, la préoccupation de robustesse devrait produire des formes de réponses encore plus variées qu'elles ne le sont actuellement. Il propose pour terminer quelques pistes de recherche pouvant précisément conduire à d'autres formes de réponses.

Les trois autres familles dont il va être question maintenant rassemblent ceux des articles qui se situent successivement sur les versants standard, mixte et concret.

Six articles se situent clairement sur le versant standard S. Il s'agit des articles de H. Aïssi, C. Bazgan, D. Vanderpooten ; V. Gabrel, C. Murat ; M. Salazar-Neumann ; V.Th. Paschos, O.A. Telelis, V. Zissimopoulos ; R. Kalaï, C. Lamboray ainsi que celui de M. Minoux.

H. Aïssi *et al.* proposent un état de l'art sur les versions min-max et min-max regret de quelques problèmes combinatoires. Les auteurs s'intéressent particulièrement aux problèmes de plus court chemin, d'arbre couvrant, d'affectation, de coupe, de (s − t)-coupe et de sac à dos. Ils présentent des résultats de complexité ainsi que des algorithmes de résolution pour les versions min-max et min-max regret de ces problèmes.

L'article de V. Gabrel et C. Murat ainsi que celui de M. Salazar-Neumann traitent le problème de plus court chemin robuste. Tandis que le premier fait un état de l'art général sur les problèmes de plus courts chemins pour lesquels il existe des éléments d'incertitude et d'indétermination sur les valeurs des arcs, le second se focalise sur la détermination des arcs strictement forts et les arcs non faibles d'un graphe orienté dans le cas particulier où l'incertitude sur les arcs est modélisée par des intervalles.

V.Th. Paschos *et al.* étudient, quant à eux, une version robuste du problème de l'arbre de Steiner. Ils utilisent un modèle d'optimisation stochastique à deux étages pour résoudre le problème dans le cas d'incertitude sur la présence de chaque sommet du graphe complet initial dans le graphe final.

R. Kalaï et C. Lamboray s'intéressent à la comparaison de deux approches de robustesse fondées sur le concept de quasi-optimalité. Les auteurs mettent en évidence quelques propriétés communes aux deux approches et démontrent que l'une d'elle peut être considérée comme une relaxation de l'autre.

Enfin, M. Minoux traite des notions de dualité et de robustesse dans les problèmes de programmation linéaire. En particulier, il démontre que la version robuste du dual d'un PL n'est pas forcément équivalente au dual de la version robuste. Il s'intéresse également à une sous-classe de problèmes linéaires robustes et introduit le concept de modèle linéaire robuste à deux étages. Pour démontrer l'intérêt pratique de ce nouveau modèle, l'auteur l'applique au problème d'ordonnancement PERT robuste.

Il nous a paru justifié de situer sur le versant mixte M les six autres articles que nous présentons brièvement ci-après.

L'article de M. Aloulou et C. Artigues ainsi que celui de E. Sanlaville traitent de robustesse en ordonnancement. Le premier considère le problème de pilotage d'atelier en temps réel. Les auteurs proposent de construire d'une façon procative une solution présentant de la flexibilité séquentielle pouvant être exploitée en temps réel pour absorber les incertitudes du modèle et les perturbations. Ceci peut être réalisé en définissant un ordre partiel des opérations au niveau de chaque machine, laissant la possibilité au décideur de compléter les décisions de séquencement en fonction de ses préférences et de l'état de l'exécution. Proposer une solution partielle n'est pertinent que si on est capable de donner une évaluation de tous les ordonnancements qui peuvent être obtenus par extension de cette solution. L'article répond alors aux deux questions suivantes : Quelle est la pire performance des ordonnancements pouvant être atteints par une règle de pilotage suivie par le décideur ? Comment construire une solution flexible avec des garanties de performance imposées ?

L'article de E. Sanlaville est avant tout un état de l'art de travaux existants incluant quelques nouveaux résultats pour le problème d'ordonnancement de deux machines parallèles identiques avec incertitudes sur les délais de communication. Deux analyses sont proposées : une analyse de sensibilité des algorithmes optimaux en contexte déterministe et une analyse de robustesse en présence d'incertitude. Les deux cas de petits et de grands délais de communication sont considérés séparément. Les études montrent le rôle central d'algorithmes processeur-ordonnés, c'est-à-dire des algorithmes qui bâtissent des affectations telles que toutes les communications se font d'une machine « émettrice » vers une machine « réceptrice ». Ce type d'analyse peut être étendu à de nombreux autres problèmes d'ordonnancement.

Deux papiers font références à un type particulier de problèmes concrets.

K. Sörensen et M. Sevaux dans lequel les auteurs explorent comment des solutions robustes et flexibles d'un certain nombre de variantes du problème de tournées de véhicules peuvent être obtenues. Dans ce but, une approche par échantillonage pour estimer la robustesse et la flexibilité des solutions est combinée avec une méta-heuristique. Cette combinaison permet de résoudre des problèmes plus grands et avec des structures stochastiques plus complexes que les méthodes traditionnelles basées sur la programmation stochastique. Les auteurs mettent en avant le fait que le risque pris par le décideur doit être pris en compte lors du choix des solutions robustes et flexibles et montrent comment ceci peut être intégré dans leur approche.

P. Vallin se propose d'exhiber des conclusions et solutions robustes dans un problème d'investissement sous contrainte budgétaire avec une incertitude se traduisant par la méconnaissance des rentabilités des investissements (ou placements). La notion de robustesse est appréhendée à travers le critère de min-max regret. L'analyse aboutit à des conclusions génériques intéressantes. Par exemple, lorsque le taux maximum de perte est supérieur au taux maximum de gain, la consommation totale du budget n'est pas une politique robuste mais, pour obtenir une politique robuste, il faut investir dans les deux placements à plus faible coût.

Enfin, deux articles traitent de procédures susceptibles d'êtres utilisées dans des contextes variés.

S. Greco *et al* présentent une nouvelle méthode, appelée $UTA^{GMS}$, pour le rangement multicritère d'un ensemble fini d'alternatives utilisant un ensemble de fonctions additives qui résultent d'une régression linéaire. $UTA^{GMS}$ utilise une information de préférence indirecte dans le processus de désagrégation et présente deux rangements sur l'ensemble des alternatives : le « rangement nécessaire » pour lequel une action *a* est au moins aussi bien classée qu'une action *b* pour toutes les fonctions additives compatibles avec l'information de préférence et le « rangement possible » pour lequel *a* est au moins aussi bien classée que *b* pour au moins une fonction additive. Le « classement nécessaire » peut alors être considéré comme une conclusion robuste donnée par la méthode.

C. Lamboray considère le problème où des rangements, fournis par exemple par des évaluateurs, doivent être agrégés en un rangement de compromis. Il propose de construire des conclusions robustes sur des ordres prudents afin de gérer la possible multiplicité des solutions de compromis. Ceci est motivé par la suggestion de Arrow et Raynaud qu'un rangement de compromis devait être un ordre prudent. L'approche proposée permet de raffiner progressivement le modèle d'aide à la décision. L'auteur illustre son approche sur un exemple de rangement de priorités de recherche par un groupe de jeunes chercheurs.

A notre grand regret aucune contribution n'a été retenue sur le versant *concret C.*

Bernard Roy, Mohamed Ali Aloulou, Rim Kalaï

# Foreword

Most of the papers in this volume were initially presented and debated within a working group created in 2003 by Bernard Roy under the aegis of LAMSADE. The goal of this group was to discuss, compare and deepen the ever more numerous and more diverse research about robustness in the context of operational research and decision aiding tools and to work towards defining a common research framework. To this end, two day-long meetings were held each year, bringing together researchers from multiple universities, primarily French and Belgian. This working group was supported financially by the French CNRS and the Belgian FNRS/CGRI.

The editors of this volume appointed two reviewers for each paper for a review process. The final articles are presented in this volume in alphabetical order, under the name of the first author. However, here in this editorial, the diverse articles have been grouped into four general families.

The first one groups together articles by L. Dias and B. Roy, who try to underline the interest of the subject of this volume, highlighting the variety of the research currently available in the literature.

Dias' article begins with a rapid presentation of the most current difficulties encountered by OR researchers in setting the parameter values of individual or group decision aiding models. The objective of robustness analysis is to respond to these difficulties. The author describes three possible roles for robustness analysis, highlighting the potential benefits of using robustness analysis as a tool to guide the entire decision process, as is the case in the VIP Analysis and IRIS software cited in the text. The author emphasizes the need to search for robust *conclusions* rather than robust *solutions*, thus underlining the role of modeling as a constructive learning process.

After defining what he means by "robust" in the OR-DA context, B. Roy explains why he prefers to speak about "robustness concerns" rather than "robustness analysis", which is likely to be interpreted too narrowly, and describes in detail the multiple reasons for these concerns. In order to better show the multi-faceted character of these concerns, he suggests grouping the studies that deal with robustness on three territories. The first two territories, called *standard* (S) and *concrete* (C), have characteristic features which oppose them in many ways. The third one, *mixed* (M), intends to support those works which do not possess all the characteristic features required for being clearly assigned to one on the two other territories. In the fourth section, he shows that, on each of the three territories, considering these robustness concerns provide even more varied responses than those currently existing. He ends by suggesting several avenues of research that would lead directly to other kinds of responses.

The remaining articles in this volume are grouped according to whether they can be situated on one of the three territories.

Six articles are clearly on the *standard* territory. They include those by H. Aïssi, C. Bazgan, D. Vanderpooten; V. Gabrel, C. Murat; M. Salazar-Neumann; V.Th. Paschos, A.O. Telelis, V. Zissimopoulos; R. Kalaï, C. Lamboray and M. Minoux.

Aïssi *et al.* survey complexity results for the min-max and min-max regret versions of some combinatorial optimization problems. The authors are particularly interested in shortest-path problems, spanning tree problems, assignment problems, cut problems, s-t cut problems, and knapsack problems. They present complexity results and algorithms to solve these problems optimally.

The article by Gabrel & Murat, as well as the one by Salazar-Neumann, deal with the robust shortest-path problem. The former present a general State-of-the-Art of shortest-path problems in which uncertainty affects the arc values. The latter focuses on strictly determining the strong and non-weak arcs in a directed graph for the special case in which uncertainty is modeled using intervals.

Paschos *et al.* present a study of a robust version of Steiner's tree problem. They use a two-stage stochastic optimization model to solve the problem for the case of uncertainty related to the presence in the final graph of the nodes constituting the initial complete weighted graph.

Kalai & Lamboray compare two approaches of robustness based on the quasi-optimality concept. The authors highlight the properties that these two approaches have in common and demonstrate that one of them can be considered to be a relaxation of the other.

Minoux tackles the notions of duality and robustness in linear programming (LP) problems. Specifically, he demonstrates the dual of a given robust linear programming model is not equivalent to the robust version of the dual. He deepens his investigations of duality in the context of robustness by considering a sub-class of robust LP problems involving uncertainty and introduces the idea of a robust 2-stage LP model. To show the practical advantages of this new model, the author applies it to a robust PERT scheduling problem.

The next six articles can be situated on the territory M.

The first two articles in this category deal with scheduling:

The article by Aloulou & Artigues, as well as the one by Sanlaville, deals with robustness in scheduling. Aloulou & Artigues consider the problem of workshop piloting in real time. The authors propose proactively building a sequentially flexible solution that can be exploited in real time, allowing both perturbations and model uncertainty to be absorbed. This can be accomplished by defining a partial order of operations for each machine, while making it possible for decision-makers to modify the sequencing decisions according to their preferences and the execution status. Still, proposing a partial solution is only useful if it is possible to evaluate all the schedules that can be obtained by extending this solution. The article thus responds to the following question: What is the worst scheduling performance that can be attained using a piloting rule monitored by the decision-maker?

Sanlaville's article examines the problem of scheduling with uncertainties. For a specific case of scheduling two identical machines with uncertainties related to delays in communication, two methods are proposed for finding a robust solution: sensitivity analysis for deterministic-optimal algorithms and robustness analysis. The cases of small and large communication delays are considered separately. The author shows the central role of the processor-ordered algorithms (i.e., the algorithms that build assignments so that all the communications go from a "sending" machine to a "receiving" machine). These two types of analysis can be extended to numerous other scheduling problems.

The second two articles deal with a particular type of concrete problem:
Sörensen & Sevaux explore how robust and flexible solutions of a number of stochastic variants of the capacitated vehicle routing problem can be obtained. To this end, they combine a sampling approach for estimating the robustness or flexibility of a solution with a meta-heuristic optimization technique. This combination allows solutions to larger problems with more complex stochastic structures than the traditional stochastic programming methods. It is also more flexible in that the approach can be easily adapted to more complex problems. The authors argue that the decision-maker's risk preference must be explicitly taken into account when choosing a robust or flexible solution and show how this can be done using their approach.

Vallin proposes conclusions and robust solutions for a investment problem with budgetary constraints and uncertainty. This uncertainty comes from not knowing the profitability of the investments on a fixed horizon. The notion of robustness is given using the min-max regret criteria. The analysis leads to interesting general conclusions. For example, when the minimum rate of loss is higher than the maximum rate of profit, the overall budget expenditure does not constitute a robust policy. In order to obtain a robust policy, it is necessary to chose either one or both of the two least expensive investments possibilities.

The last two articles deal with procedures that can be used in a variety of contexts:

Greco *et al.* present a new method, called $UTA^{GMS}$, for the multi-criteria ranking of a finite set of alternatives, using a set of additive value functions produced by ordinal regression. $UTA^{GMS}$ uses indirect preference information in the disaggregation process and presents two possible rankings of the alternatives set: the "necessary" ranking for which an action *a* is at least as well ranked as an action *b* for all the additive value functions compatible with the preference information, and the "possible" ranking for which action *a* is at least as well ranked as action *b* for at least one additive value function. The "necessary" ranking can thus be considered a robust conclusion given by the method.

Lamboray considers the problem in which various rankings (e.g., those provided by evaluators) must be aggregated in a compromise ranking. He proposes constructing robust conclusions on prudent orders in order to manage the possible multiplicity of the compromise solutions. This idea is based on Arrow & Raynaud's suggestion that a compromise ranking must be a prudent order. The proposed approach permits the decision-making model to be refined progressively. The author uses the example of a group of young researchers organizing their research priorities to illustrate his approach.

Unfortunately, no article situated on the *concrete* territory C was retained for publication.

Bernard Roy, Mohamed Ali Aloulou, Rim Kalaï

## Sommaire/Contents

# Min-max and min-max regret versions of some combinatorial optimization problems : a survey

Hassene Aissi*, Cristina Bazgan*, and Daniel Vanderpooten*

**Résumé**

Les critères min-max et min-max regret sont souvent utilisés en vue d'obtenir des solutions robustes. Après avoir motivé l'utilisation de ces critères, nous présentons des résultats généraux. Ensuite, nous donnons des résultats de complexité des versions min-max et min-max regret de quelques problèmes d'optimisation combinatoire : plus court chemin, arbre couvrant, affectation, coupe, $s - t$ coupe, sac à dos. Comme la plupart de ces problèmes sont *NP*-difficiles, nous présentons des algorithmes d'approximation ainsi que des algorithmes exacts de résolution.

**Mots-clefs :** Min-max, min-max regret, optimisation combinatoire, complexité, approximation, analyse de robustesse.

**Abstract**

Min-max and min-max regret criteria are commonly used to define robust solutions. After motivating the use of these criteria, we present general results. Then, we survey complexity results for the min-max and min-max regret versions of some combinatorial optimization problems: shortest path, spanning tree, assignment, cut, s-t cut, knapsack. Since most of these problems are *NP*-hard, we also investigate the approximability of these problems. Furthermore, we present algorithms to solve these problems to optimality.

**Key words :** Min-max, min-max regret, combinatorial optimization, complexity, approximation, robustness analysis.

*LAMSADE, Université Paris-Dauphine, 75775 Paris cedex 16, France. {aissi,bazgan,vdp}@lamsade.dauphine.fr

# 1 Introduction

The definition of an instance of a combinatorial optimization problem requires to specify parameters, in particular coefficients of the objective function, which may be uncertain or imprecise. Uncertainty/imprecision can be structured through the concept of *scenario* which corresponds to an assignment of plausible values to model parameters. There exist two natural ways of describing the set of all possible scenarios. In the *discrete scenario case*, the scenario set is described explicitly. In the *interval scenario case*, each numerical parameter can take any value between a lower and an upper bound. The min-max and min-max regret criteria, stemming from decision theory, are often used to obtain solutions hedging against parameters variations. The min-max criterion aims at constructing solutions having the best possible performance in the worst case. The min-max regret criterion, less conservative, aims at obtaining a solution minimizing the maximum deviation, over all possible scenarios, between the value of the solution and the optimal value of the corresponding scenario.

The study of these criteria is motivated by practical applications where an anticipation of the worst case is crucial. For instance, consider the sensor placement problem when designing contaminant warning systems for water distribution networks [16, 47]. A key deployment issue is identifying where the sensors should be placed in order to maximize the level of protection. Quantifying the protection level using the expected impact of a contamination event is not completely satisfactory since the issue of how to guard against potentially high-impact events, with low probability, is only partially handled. Consequently, standard approaches, such as deterministic or stochastic approaches, will fail to protect against exceptional high-impact events (earthquakes, hurricanes, 9/11-style attacks,…). In addition, reliable estimation of contamination event probabilities is extremely difficult. Min-max and min-max regret criteria are appropriate in this context by focusing on a subset of high-impact contamination events, and placing sensors to minimize the impact of such events.

The purpose of this paper is to review the existing literature on the min-max and min-max regret versions of combinatorial optimization problems with an emphasis on the complexity, the approximation and the exact resolution of these problems, both for the discrete and interval scenario cases.

The rest of the paper is organized as follows. Section 2 introduces, illustrates and motivates the min-max and min-max regret criteria. General results for these two criteria are presented in section 3. Section 4 provides complexity results for the min-max and min-max regret versions of various combinatorial optimization problems. Since most of these problems are *NP*-hard, section 5 describes the approximability of these problems. Section 6 describes exact procedures to solve min-max and min-max regret versions in the discrete and interval scenario cases. Conclusions are provided in the final section.

# 2 Presentation and motivations

We consider in this paper the class $\mathcal{C}$ of 0-1 problems with a linear objective function defined as:

$$\begin{cases} \min(\text{or } \max) \sum_{i=1}^{n} c_i x_i & c_i \in \mathbb{N} \\ x \in X \subset \{0,1\}^n \end{cases}$$

This class encompasses a large variety of classical combinatorial problems, some of which are polynomial-time solvable (shortest path problem, minimum spanning tree, ...) and others are *NP*-difficult (knapsack, set covering, ...).

In the following, all the definitions and results are presented for minimization problems $\mathcal{P} \in \mathcal{C}$. In general, they also hold with minor modifications for maximization problems, except for some cases that will be explicitly mentioned.

## 2.1 Min-max, min-max regret versions

In the discrete scenario case, the min-max or min-max regret version associated to a minimization problem $\mathcal{P} \in \mathcal{C}$ has as input a finite set $S$ of scenarios where each scenario $s \in S$ is represented by a vector $c^s = (c_1^s, \ldots, c_n^s)$.

In the interval scenario case, each coefficient $c_i$ can take any value in the interval $[\underline{c}_i, \overline{c}_i]$. In this case, the scenario set $S$ is the cartesian product of the intervals $[\underline{c}_i, \overline{c}_i]$, $i = 1, \ldots, n$. An *extreme scenario* is a scenario where $c_i = \underline{c}_i$ or $\overline{c}_i$, $i = 1, \ldots, n$.

We denote by $val(x, s) = \sum_{i=1}^{n} c_i^s x_i$ the value of solution $x \in X$ under scenario $s \in S$, by $x_s^*$ an optimal solution under scenario $s$, and by $val_s^* = val(x_s^*, s)$ the corresponding optimal value.

The min-max version corresponding to $\mathcal{P}$ consists of finding a solution having the best worst case value across all scenarios, which can be stated as:

$$\min_{x \in X} \max_{s \in S} val(x, s) \tag{1}$$

This version is denoted by DISCRETE MIN-MAX $\mathcal{P}$ in the discrete scenario case, and by INTERVAL MIN-MAX $\mathcal{P}$ in the interval scenario case.

Given a solution $x \in X$, its *regret*, $R(x, s)$, under scenario $s \in S$ is defined as $R(x, s) = val(x, s) - val_s^*$. The *maximum regret* $R_{max}(x)$ of solution $x$ is then defined as $R_{max}(x) = \max_{s \in S} R(x, s)$.

The min-max regret version corresponding to $\mathcal{P}$ consists of finding a solution minimizing its maximum regret, which can be stated as:

$$\min_{x \in X} R_{max}(x) = \min_{x \in X} \max_{s \in S} (val(x, s) - val_s^*) \tag{2}$$

This version is denoted by DISCRETE MIN-MAX REGRET $\mathcal{P}$ in the discrete scenario case, and by INTERVAL MIN-MAX REGRET $\mathcal{P}$ in the interval scenario case.

For maximization problems, corresponding max-min and min-max regret versions can be defined.

## 2.2 An illustrative example

In order to illustrate the previous definitions and to show the interest of using the min-max and min-max regret criteria, we consider a capital budgeting problem with uncertainty or imprecision on the expected profits. Suppose we wish to invest a capital $b$ and we have identified $n$ investment opportunities. Investment $i$ requires a cash outflow $w_i$ at present time and yields an expected profit $p_i$ at some term, $i = 1, \ldots, n$. Due to various exogenous factors (evolution of the market, inflation conditions, ...), the profits are evaluated with uncertainty/imprecision.

The capital budgeting problem is a knapsack problem where we look for a subset $I \subseteq \{1, \ldots, n\}$ of items such that $\sum_{i \in I} w_i \leq b$ which maximizes the total expected profit, i.e., $\sum_{i \in I} p_i$.

Depending on the type of uncertainty or imprecision, we use either discrete or interval scenarios. When the profits of investments are influenced by the occurrence of well-defined future events (e.g., different levels of market reactions, public decisions of constructing or not a facility which would impact on the investment projects), it is natural to define discrete scenarios corresponding to each event. When the evaluation of the profit is simply imprecise, a definition using interval scenarios is more appropriate.

As an illustration, consider a small size capital budgeting problem where profit uncertainty or imprecision is modelled with three discrete scenarios or with interval scenarios. Numerical values are given in Table 1 with $b = 12$ and $n = 6$.

| $i$ | $w_i$ | $p_i^1$ | $p_i^2$ | $p_i^3$ | $\underline{p}_i$ | $\overline{p}_i$ |
|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 3 | 3 | 3 | 5 |
| 2 | 5 | 8 | 4 | 6 | 2 | 6 |
| 3 | 2 | 5 | 3 | 3 | 2 | 5 |
| 4 | 4 | 3 | 2 | 4 | 2 | 3 |
| 5 | 5 | 2 | 8 | 2 | 3 | 9 |
| 6 | 3 | 4 | 6 | 2 | 1 | 7 |

Table 1: Cash outflows and profits of the investments

| 1 | 2 | 3 | Optimal solution |
|---|---|---|---|
| 17 | 10 | 12 | scenario 1: (1,1,1,0,0,0) |
| 11 | 17 | 7 | scenario 2: (0,0,1,0,1,1) |
| 16 | 9 | 13 | scenario 3: (0,1,1,1,0,0) |
| 15 | 12 | 12 | max-min: (0,1,0,1,0,1) |
| 15 | 15 | 11 | min-max regret: (0,1,1,0,1,0) |

Table 2: Optimal values and solutions (discrete scenario case)

In the discrete scenario case, we observe in Table 2 that the optimal solutions of the three scenarios are not completely satisfactory since they give low profits in some scenarios. An appropriate solution should behave well under any variation of the future profits. In this example, optimal solutions to max-min and min-max regret versions are more acceptable solutions since their performances are more stable. A risk-averse decision maker would favor the max-min solution that guarantees at least 12 in all scenarios. If the decision maker is willing to accept a small degradation of the performance in the worst case scenario with an increase of the average performance over all scenarios, then the optimal solution of the min-max regret version is more appropriate than the optimal solution of the min-max version.

In the interval scenario case, the optimal solution to the max-min version is obtained by considering only the worst-case scenario (see section 3.2) and corresponds to $x_1 = x_3 = x_5 = 1$; $x_2 = x_4 = x_6 = 0$ with a worst value 8. The obtained solution only depends on one scenario and, in particular, is completely independent of the interval upper bound values. An optimal solution of the min-max regret version is given by $x_1 = x_5 = x_6 = 1$; $x_2 = x_3 = x_4 = 0$ with a worst value and a maximum regret 7. Unlike for the max-min criterion, the optimal min-max regret solution does depend on the interval lower and upper bound values. In particular, when an interval upper bound increases, the maximum regret of each feasible solution either increases or remains stable. Obviously, this may impact on the resulting optimal solution. For instance, if we increase $\overline{p}_3$ from 5 to 8, the new min-max regret optimal solution becomes $x_3 = x_5 = x_6 = 1$; $x_1 = x_2 = x_4 = 0$ with a worst value 6 and a maximum regret 8. Observe, in the previous example, that improving the prospects of investment 3 leads to include this investment in the new optimal solution.

## 2.3 Relevance and limits of the min-max (regret) criteria

When uncertainty or imprecision on the parameter values of decision aiding models is a crucial issue, decision makers may not feel confident using results derived from parameters taking precise values. *Robustness analysis* is a theoretical framework that enables the decision maker to take into account uncertainty or imprecision in order to produce decisions that will behave reasonably under any likely input data (see, e.g., Roy [42], Vincke [45] for general contexts, Kouvelis and Yu [30] in combinatorial optimization, and Mulvey, Verderbel, and Zenios [39] in mathematical programming). Different criteria can be used to select among robust decisions. The min-max and min-max regret criteria are often used to obtain conservative decisions to hedge against variations of the input data [41]. We briefly review situations where each criterion is appropriate.

The min-max criterion is suited for non-repetitive decisions (construction of a high voltage line, highway, ...) and for decision environments where precautionary measures are needed (nuclear accidents, public health). In such cases, classical approaches like deterministic or stochastic optimization are not relevant. This criterion is also appropriate in competitive situations [41] or when the decision maker must reach a pre-defined goal (sales, inventory, ...) under any variation of the input data.

The min-max regret criterion is suitable in situations where the decision maker may feel regret if he/she makes a wrong decision. He/she thus takes this anticipated regret into account when deciding. For instance, in finance, an investor may observe not only his own portfolio performance but also returns on other stocks or portfolios in which he was able to invest but decided not to [20]. Therefore, it seems very natural to assume that the investor may feel joy/disappointment if his own portfolio outperformed/underperformed some benchmark portfolio or portfolios. The min-max regret criterion reflects such a behavior and is less conservative than the min-max criterion since it considers missed opportunities. It has also been shown that, in the presence of uncertainty on prices and yields, the behavior of some economical agents (farmers) could sometimes be better predicted using a min-max regret criterion, rather than a classical profit maximization criterion [29].

Maximum regret can also serve as an indicator of how much the performance of a decision can be improved if all uncertainties/imprecisions could be resolved [30]. The min-max regret criterion is relevant in situations where the decision maker is evaluated ex-post.

Another justification of the min-max regret criterion is as follows. In uncertain/imprecise decision contexts, a reasonable objective is to find a solution with performances as close as possible from the optimal values under all scenarios. This amounts to setting a threshold $\varepsilon$ and looking for a solution $x \in X$ such that $val(x,s) - val_s^* \leq \varepsilon$, for all $s \in S$. Equivalently, $x$ should satisfy $R_{max}(x) \leq \varepsilon$. Then, looking for such a solution with $\varepsilon$ as small as possible is equivalent to determining a min-max regret solution.

Min-max and min-max regret criteria are simple to use since they do not require any additional information unlike other approaches based on probability or fuzzy set theory. Furthermore, they are often considered as reference criteria and a starting point in robustness analysis. However, the min-max and min-max regret criteria are sometimes inappropriate. In some situations, these criteria are too pessimistic for decision makers who are willing to accept some degree of risk. In addition, they attach a great importance to worst case scenarios which are sometimes unlikely to occur. In the discrete scenario case, this difficulty could be handled, at the modelling stage, by including only relevant scenarios in the scenario set. In the interval scenario case, min-max and min-max regret optimal solutions are obtained for extreme scenarios, as will be shown in section 3.2. Clearly not all these scenarios are likely to happen and this limits the applicability of these criteria. This difficulty becomes all the more important than the intervals get larger.

Some approaches have been designed so as to reduce these drawbacks. In the discrete scenario case, Daskin, Hesse, and ReVelle [19] propose a model called the $\alpha$-reliable min-max regret model that identifies a solution that minimizes the maximum regret with respect to a selected subset of scenarios whose probability of occurrence is at least some user-defined value $\alpha$. Kalai, Aloulou, Vallin, and Vanderpooten [25] propose another approach called lexicographic $\alpha$-robustness which, instead of focusing on the worst case, considers all scenarios in lexicographic order, from the worst to the best, and also includes a tolerance threshold $\alpha$ in order not to discriminate among solutions with similar values. In the interval scenario case, Bertsimas and Sim [15] propose a robustness model with a scenario set restricted to scenarios where the number of parameters that are pushed to their upper bounds is limited by a user-specified parameter. The robust version obtained using this model has the same approximation complexity as the classical version. This is a clear advantage over the classical robustness criteria since their robust counterpart are generally *NP*-hard. The main difficulty is, however, the definition of this technical parameter whose value may impact on the resulting solution.

# 3 General results on min-max (regret) versions

We present in this section general results that give an insight on the nature and difficulty of min-max (regret) versions. They also serve as a basis for further results or algorithms presented in the next sections.

## 3.1 Discrete scenario case

### 3.1.1 Relationships between min-max (regret) and classical versions

We investigate, in this part, the quality of the solutions obtained by solving $\mathcal{P}$ for specific scenarios.

A first attempt to solve DISCRETE MIN-MAX (REGRET) $\mathcal{P}$ is to construct an optimal solution $x_s^*$ for each scenario $s \in S$, compute $\max_{s \in S} val(x_s^*, s)$ (or $R_{max}(x_s^*)$), and then, select the best of the obtained candidate solutions. However, this solution can be very bad, since the gap between its value and the optimal value may increase exponentially in the size of the instance as we can see in the following example. Consider for example a pathological instance of DISCRETE MIN-MAX (REGRET) SHORTEST PATH. Figure 1 depicts a graph $G = (V, A)$ with $2n$ vertices where each arc is valued by costs on 2 scenarios. The optimal solution in the first scenario is given by path $1, n+1, \ldots, 2n-1, 2n$ with values $(0, 2^n - 1)$, a worst value $2^n - 1$ and a maximum regret $2^n - 1$, whereas the optimal solution in the second scenario is given by path $1, 2, \ldots, n, 2n$ with values $(2^n - 1, 0)$, a worst value $2^n - 1$, and a maximum regret $2^n - 1$. However, the optimal solution of the min-max and min-max regret versions is given by path $1, 2n$ with a worst value 1 and a maximum regret 1. Thus the gap between the best value (regret) among the optimal solutions of each scenario and the optimum of min-max (min-max regret) is $2^n - 2$.

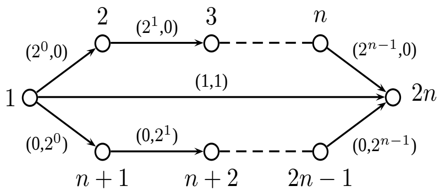



Figure 1: A critical instance of DISCRETE MIN-MAX (REGRET) SHORTEST PATH where optimal solutions in each scenario are very bad

The previous example shows that by solving $\mathcal{P}$ for all scenarios in *S*, we cannot guarantee the performance of the obtained solution. Another idea is to solve problem $\mathcal{P}$ for a fictitious scenario in the hope that the obtained solution has a good performance. The following proposition considers the *median* scenario, i.e. with costs corresponding to the average value over all scenarios.

**Proposition 1 ([30])** *Consider an instance $I$ of* DISCRETE MIN-MAX $\mathcal{P}$ *(or* DISCRETE MIN-MAX REGRET $\mathcal{P}$*) with $k$ scenarios where each scenario $s \in S$ is represented by $(c_1^s, \ldots, c_n^s)$ and where $\mathcal{P}$ is a minimization problem. Consider also an instance $I'$ of $\mathcal{P}$ where each coefficient of the objective function is defined by $c_i' = \sum_{s=1}^{k} \frac{c_i^s}{k}$, $i = 1, \ldots, n$. Then $x'$ an optimal solution of $I'$ is such that* $\max_{s \in S} val(x', s) \leq k.opt(I)$ *(or $R_{max}(x') \leq k.opt(I)$) where $opt(I)$ denotes the optimal value of instance $I$.*

**Proof:** Consider an instance $I$ of DISCRETE MIN-MAX REGRET $\mathcal{P}$. Let $x'$ be an optimal solution of $I'$. We first show that $L = \sum_{s \in S} \frac{1}{k}(val(x', s) - val_s^*)$ is a lower bound of $opt(I)$.

$$L = \min_{x \in X} \frac{1}{k} \sum_{s \in S} (val(x, s) - val_s^*) \leq \min_{x \in X} \frac{1}{k} k \max_{s \in S} (val(x, s) - val_s^*) = opt(I)$$

Moreover, $U = \max_{s \in S}(val(x', s) - val^*_s)$ is clearly an upper bound of $opt(I)$ and we get

$$\min_{x \in X} \max_{s \in S}(val(x, s) - val^*_s) \leq \max_{s \in S}(val(x', s) - val^*_s) \leq \sum_{s \in S}(val(x', s) - val^*_s)$$

$$= kL \leq k.opt(I)$$

The proof for DISCRETE MIN-MAX $\mathcal{P}$ is exactly the same except that we take $L = \sum_{s \in S} \frac{1}{k} val(x', s)$ and $U = \max_{s \in S} val(x', s)$ as lower and upper bounds of $opt(I)$, and we remove $val^*_s$ everywhere in the proof. □

Considering a maximization problem $\mathcal{P}$, we can define lower and upper bounds $L$ and $U$ similarly. For the min-max regret version, we also obtain a $k$-approximate solution in this way. However, the previous result does not hold for the max-min version since the ratio $\frac{U}{L}$ is not bounded. In particular, for the knapsack problem, this ratio can be exponential in the size of the input, as noticed in [49].

An alternative idea is to solve problem $\mathcal{P}$ on a fictitious scenario, called *pessimistic* scenario, with costs corresponding to the worst values over all scenarios.

**Proposition 2** *Consider an instance $I$ of* DISCRETE MIN-MAX $\mathcal{P}$ *with $k$ scenarios where each scenario $s \in S$ is represented by $(c^s_1, \ldots, c^s_n)$ and where $\mathcal{P}$ is a minimization problem. Consider also an instance $I'$ of $\mathcal{P}$ where each coefficient of the objective function is defined by $c'_i = \max_{s \in S} c^s_i$, $i = 1, \ldots, n$. Then $x'$ an optimal solution of $I'$ is such that $\max_{s \in S} val(x', s) \leq k.opt(I)$ where $opt(I)$ denotes the optimal value of instance $I$.*

**Proof :** Let $x^*$ denote an optimal solution of instance $I$. We have

$$\max_{s \in S} \sum_{i=1}^{n} c^s_i x'_i \leq \sum_{i=1}^{n} c'_i x'_i \leq \sum_{i=1}^{n} c'_i x^*_i \leq \sum_{i=1}^{n} \sum_{s \in S} c^s_i x^*_i$$

$$= \sum_{s \in S} \sum_{i=1}^{n} c^s_i x^*_i \leq k. \max_{s \in S} \sum_{i=1}^{n} c^s_i x^*_i = k.opt(I)$$

□

The same result cannot be extended to the min-max regret version. Figure 2 illustrates a critical instance of DISCRETE MIN-MAX REGRET SHORTEST PATH with two scenarios. The optimal solution for the first and second scenario is the same and corresponds to path $1, n-1, n$. Thus, this path is the min-max regret optimal solution with a maximum regret 0. However, the optimal solution for the pessimistic scenario is given by path $1, 2, \ldots, n$ with maximum regret $2^n$.

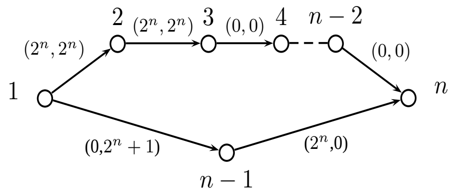



Figure 2: A critical instance of DISCRETE MIN-MAX REGRET SHORTEST PATH where the optimal solution in the pessimistic scenario is very bad

Extension of Proposition 2 to a maximization problem $\mathcal{P}$ does not hold neither for DISCRETE MAX-MIN $\mathcal{P}$ nor for DISCRETE MIN-MAX REGRET $\mathcal{P}$.

Quite interestingly, however, solving $\mathcal{P}$ for the pessimistic scenario leads to optimal solutions for the specific class of bottleneck problems defined as

$$\begin{cases} \min \max_{i=1,\dots,n} c_i x_i & c_i \in \mathbb{N} \\ x \in X \subset \{0,1\}^n \end{cases}$$

This class encompasses bottleneck versions of classical optimization problems (shortest path, spanning tree, ...). The min-max and min-max regret bottleneck versions are defined by (1) and (2) respectively, with $val(x,s) = \max_{i=1,\dots,n} c_i^s x_i$.

These versions are denoted DISCRETE MIN-MAX (REGRET) BOTTLENECK $\mathcal{P}$ in the discrete scenario case, and INTERVAL MIN-MAX (REGRET) BOTTLENECK $\mathcal{P}$ in the interval scenario case.

**Proposition 3** *Consider an instance $I$ of* DISCRETE MIN-MAX BOTTLENECK $\mathcal{P}$ *with $k$ scenarios where each scenario $s \in S$ is represented by $(c_1^s, \dots, c_n^s)$ and where $\mathcal{P}$ is a minimization problem. Consider also an instance $I'$ of $\mathcal{P}$ where each coefficient of the objective function is defined by $c_i' = \max_{s \in S} c_i^s$, $i = 1, \dots, n$. Then $x'$ an optimal solution of $I'$ is also an optimal solution of $I$.*

**Proof :**

$$\min_{x \in X} \max_{s \in S} val(x,s) = \min_{x \in X} \max_{s \in S} \max_{i=1,\dots,n} c_i^s x_i = \min_{x \in X} \max_{i=1,\dots,n} \max_{s \in S} c_i^s x_i = \min_{x \in X} \max_{i=1,\dots,n} c_i' x_i$$

□

The next proposition gives a reduction from DISCRETE MIN-MAX REGRET BOTTLENECK $\mathcal{P}$ to DISCRETE MIN-MAX BOTTLENECK $\mathcal{P}$.

**Proposition 4** *Consider an instance $I$ of* DISCRETE MIN-MAX REGRET BOTTLENECK $\mathcal{P}$ *with $k$ scenarios where each scenario $s \in S$ is represented by $(c_1^s, \dots, c_n^s)$ and where $\mathcal{P}$ is a minimization problem. Consider also an instance $\widetilde{I}$ of* DISCRETE MIN-MAX BOTTLENECK $\mathcal{P}$ *where each coefficient of the objective function is defined by $\widetilde{c}_i^s = \max\{c_i^s - val_s^*, 0\}$, $i = 1, \dots, n$. Then $\widetilde{x}$, an optimal solution of $\widetilde{I}$, is also an optimal solution of $I$.*

**Proof :**

$$\min_{x \in X} \max_{s \in S} (val(x,s) - val_s^*) = \min_{x \in X} \max_{s \in S} (\max_{i=1,\dots,n} c_i^s x_i - val_s^*) = \min_{x \in X} \max_{s \in S} \max_{i=1,\dots,n} \widetilde{c}_i^s x_i$$

□

The previous propositions are summarized in the following corollary.

**Corollary 1** *Given $\mathcal{P}$ an optimization problem,* DISCRETE MIN-MAX BOTTLENECK $\mathcal{P}$ *and* DISCRETE MIN-MAX REGRET BOTTLENECK $\mathcal{P}$ *reduce to $\mathcal{P}$.*

### 3.1.2 Relationships between min-max (regret) and multi-objective versions

It is natural to consider scenarios as objective functions. This leads us to investigate relationships between min-max (regret) and multi-objective versions.

The multi-objective version associated to $\mathcal{P} \in \mathcal{C}$, denoted by MULTI-OBJECTIVE $\mathcal{P}$, has for input $k$ objective functions where the $h$th objective function has coefficients $c_1^h, \dots, c_n^h$. We denote by $val(x,h) = \sum_{i=1}^n c_i^h x_i$ the value of solution $x \in X$ on criterion $h$, and assume w.l.o.g. that all criteria are to be minimized. Given two feasible solutions $x$ and $y$, we say that $x$ dominates $y$ if $val(x,h) \leq val(y,h)$ for $h = 1, \dots, k$ with at least one strict inequality. The problem consists of finding the set $E$ of efficient solutions. A feasible solution $x$ is *efficient* if there is no other feasible solution $y$ that dominates $x$. In general MULTI-OBJECTIVE $\mathcal{P}$ is intractable in the sense that it admits instances for which the size of $E$ is exponential in the size of the input (see, e.g., Ehrgott [21]).

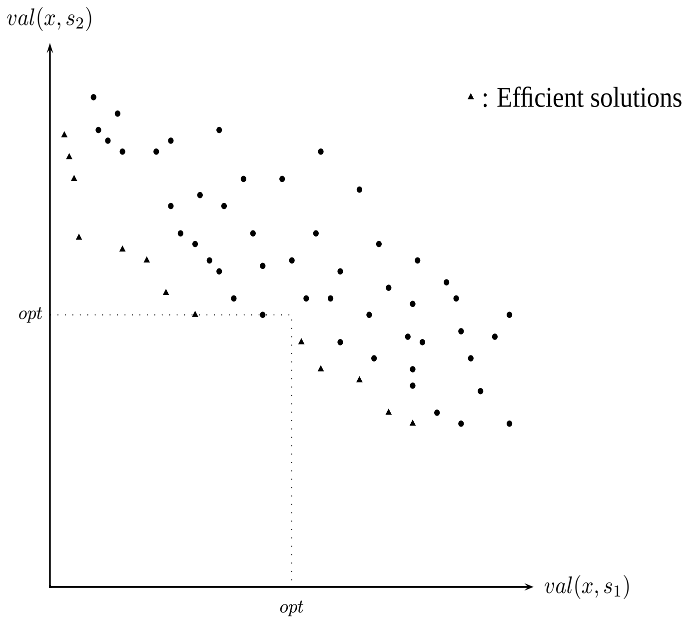



Figure 3: Relationships between min-max and multi-objective versions

**Proposition 5** *Given a minimization problem $\mathcal{P}$, at least one optimal solution for* DISCRETE MIN-MAX $\mathcal{P}$ *is necessarily an efficient solution.*

**Proof:** If $x \in X$ dominates $y \in X$ then $\max_{s \in S} val(x, s) \leq \max_{s \in S} val(y, s)$. Therefore, we obtain an optimal solution for DISCRETE MIN-MAX $\mathcal{P}$ by taking, among the efficient solutions, one that has a minimum $\max_{s \in S} val(x, s)$, see Figure 3. □

**Proposition 6** *Given a minimization problem $\mathcal{P}$, at least one optimal solution for* DISCRETE MIN-MAX REGRET $\mathcal{P}$ *is necessarily an efficient solution.*

**Proof:** If $x \in X$ dominates $y \in X$ then $val(x, s) \leq val(y, s)$, for each $s \in S$, and thus $R_{max}(x) \leq R_{max}(y)$. Therefore, we obtain an optimal solution for DISCRETE MIN-MAX REGRET $\mathcal{P}$ by taking, among the efficient solutions, a solution $x$ that has a minimum $R_{max}(x)$, see Figure 4. □

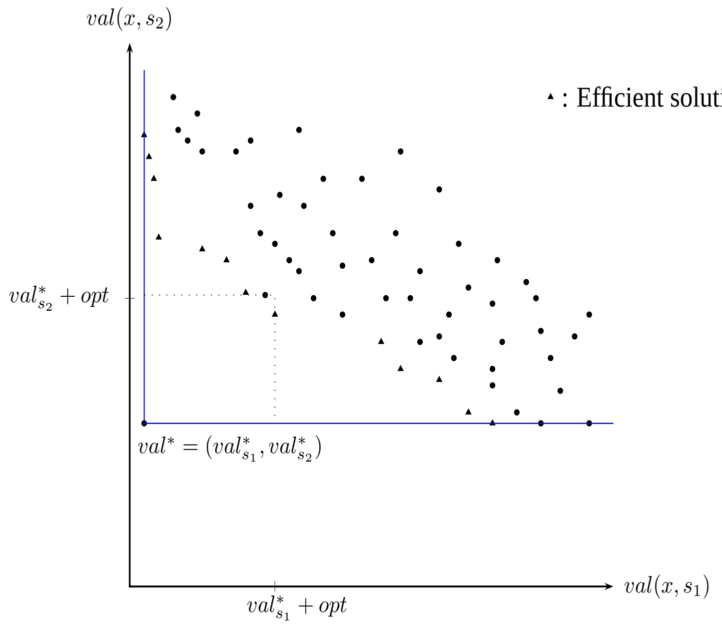



Figure 4: Relationships between min-max regret and multi-objective versions

Observe that if DISCRETE MIN-MAX (REGRET) $\mathcal{P}$ admit several optimal solutions, some of them may not be efficient, but at least one is efficient (see Figures 3 and 4).

## 3.2 Interval scenario case

For a minimization problem $\mathcal{P} \in \mathcal{C}$, solving INTERVAL MIN-MAX $\mathcal{P}$ is equivalent to solving $\mathcal{P}$ on the scenario $\overline{c} = (\overline{c}_1, \ldots, \overline{c}_n)$ where the values of all coefficients $c_i$ are set to their upper bounds (for a maximization problem, consider only scenario $\underline{c} = (\underline{c}_1, \ldots, \underline{c}_n)$). Therefore, the rest of the section is devoted to INTERVAL MIN-MAX REGRET $\mathcal{P}$.

In order to compute the maximum regret of a solution $x \in X$, we only need to consider its worst scenario $c^-(x)$. Yaman, Karaşan and Pinar [48] propose an approach to construct this scenario for INTERVAL MIN-MAX REGRET SPANNING TREE and characterize the optimal solution. These approaches can, however, be generalized to any problem $\mathcal{P} \in \mathcal{C}$.

**Proposition 7** *Given a minimization problem $\mathcal{P}$, the regret of a solution $x \in X$ is maximized for scenario $c^-(x)$ defined as follows:*

$$c_i^-(x) = \begin{cases} \overline{c}_i & \text{if } x_i = 1, \\ \underline{c}_i & \text{if } x_i = 0, \end{cases} \qquad i = 1, \ldots, n$$

**Proof:** Given a solution $x \in X$, let $I(x) = \{i \in 1, \dots, n : x_i = 1\}$. For any scenario $s \in S$, we have

$$\begin{aligned} R(x,s) = val(x,s) - val(x^*_s, s) &= \sum_{i \in I(x) \setminus I(x^*_s)} c^s_i - \sum_{i \in I(x^*_s) \setminus I(x)} c^s_i \\ &\leq \sum_{i \in I(x) \setminus I(x^*_s)} \overline{c}_i - \sum_{i \in I(x^*_s) \setminus I(x)} \underline{c}_i \\ &= val(x, c^-(x)) - val(x^*_s, c^-(x)) \\ &\leq val(x, c^-(x)) - val(x^*_{c^-(x)}, c^-(x)) \\ &= R(x, c^-(x)) \end{aligned}$$

□

The following proposition shows that an optimal solution of INTERVAL MIN-MAX REGRET $\mathcal{P}$ is an optimal solution of $\mathcal{P}$ for one of the extreme scenarios.

**Proposition 8** *Given a minimization problem $\mathcal{P}$, an optimal solution $x^*$ of* INTERVAL MIN-MAX REGRET $\mathcal{P}$ *corresponds to an optimal solution of $\mathcal{P}$ for at least one extreme scenario, in particular its most favorable scenario $c^+(x^*)$ defined as follows:*

$$c^+_i(x^*) = \begin{cases} \underline{c}_i & \text{if } x^*_i = 1, \\ \overline{c}_i & \text{if } x^*_i = 0, \end{cases} \qquad i = 1, \dots, n$$

**Proof:** Let $x^*$ denote an optimal solution of INTERVAL MIN-MAX REGRET $\mathcal{P}$ and $I(x) = \{i \in 1, \dots, n : x_i = 1\}$. For any $s \in S$, we have

$$\begin{aligned} val(x^*, s) - val(x^*_{c^+(x^*)}, s) = & \sum_{i \in I(x^*) \setminus I(x^*_{c^+(x^*)})} c^s_i - \sum_{i \in I(x^*_{c^+(x^*)}) \setminus I(x^*)} c^s_i \\ \geq & \sum_{i \in I(x^*) \setminus I(x^*_{c^+(x^*)})} \underline{c}_i - \sum_{i \in I(x^*_{c^+(x^*)}) \setminus I(x^*)} \overline{c}_i \\ = & \; val(x^*, c^+(x^*)) - val(x^*_{c^+(x^*)}, c^+(x^*)) \end{aligned}$$

Suppose that $x^*$ is not an optimal solution of $\mathcal{P}$ for its most favorable scenario $c^+(x^*)$. Then the previous expression is strictly positive. Consequently, we have

$$val(x^*, s) - val^*_s > val(x^*_{c^+(x^*)}, s) - val^*_s, \text{ for all } s \in S$$

thus

$$\max_{s \in S} \{val(x^*, s) - val^*_s\} > \max_{s \in S} \{val(x^*_{c^+(x^*)}, s) - val^*_s\}$$

This contradicts the optimality of $x^*$ for INTERVAL MIN-MAX REGRET $\mathcal{P}$. □

The two previous propositions suggest the following direct procedure to solve exactly the min-max regret version. First construct an optimal solution $x_s^*$ for each extreme scenario $s$ and compute $R_{max}(x_s^*)$ using Proposition 7. Then, select among these solutions one having the minimum maximum regret. This procedure is in general impracticable since the number of extreme scenarios is up to $2^n$. Nevertheless, if the number $u \leq n$ of uncertain/imprecise coefficients $c_i$, corresponding to non degenerate intervals, is constant or bounded by the logarithm of a polynomial function in the size of the input, then INTERVAL MIN-MAX REGRET $\mathcal{P}$ has the same complexity as $\mathcal{P}$, as noticed by Averbakh and Lebedev [14].

As for the discrete scenario case, we briefly investigate the quality of the solutions obtained by solving $\mathcal{P}$ on a specific scenario.

A first idea, is to consider the worst scenario $\overline{c} = (\overline{c}_1, \ldots, \overline{c}_n)$, that is to take an optimal min-max solution as a candidate solution for the min-max regret version.

An optimal solution of INTERVAL MIN-MAX $\mathcal{P}$ has, in practice, a maximum regret close to the optimal value of the min-max regret version but, in the worst case, its performance is very bad. Figure 5 presents a critical example for SHORTEST PATH. The optimal solution of the min-max version is given by path $1, n$ with a worst value and a maximum regret $2^n$. On the other hand, the optimal solution of the min-max regret version is given by path $1, 2, \ldots, n$ with a worst value $2^n + 1$ and a maximum regret 1.

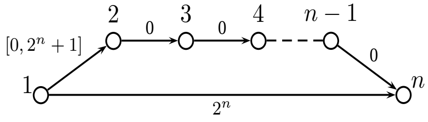



Figure 5: A critical instance of INTERVAL MIN-MAX REGRET SHORTEST PATH where the optimal min-max solution is very bad

Kasperski and Zieliński [28] show that, by solving problem $\mathcal{P}$ on a particular scenario, we can obtain a good upper bound of the optimal value of INTERVAL MIN-MAX REGRET $\mathcal{P}$.

**Proposition 9 ([28])** *Given an instance $I$ of* INTERVAL MIN-MAX REGRET $\mathcal{P}$, *consider an instance $I'$ of $\mathcal{P}$ where each coefficient of the objective function is defined by $c'_i = \frac{1}{2}(\underline{c}_i + \overline{c}_i)$. Then $x'$ an optimal solution of $I'$ has $R_{max}(x') \leq 2opt(I)$ where $opt(I)$ denotes the optimal value of instance $I$.*

For the class of bottleneck problems, as for the discrete scenario case (see Propositions 3 and 4), an optimal solution of INTERVAL MIN-MAX REGRET BOTTLENECK $\mathcal{P}$ can be obtained by solving BOTTLENECK $\mathcal{P}$ on a carefully selected scenario. Let $s_i$ denote the scenario where $c_i = \overline{c}_i$ and $c_j = \underline{c}_j$ for $j \neq i$, and $i = 1, \ldots, n$.

**Proposition 10 ([9])** *Given an instance $I$ of* INTERVAL MIN-MAX REGRET BOTTLENECK $\mathcal{P}$*, consider an instance $I'$ of* BOTTLENECK $\mathcal{P}$ *where each coefficient of the objective function is defined by $c'_i = \max\{\overline{c}_i - val^*_{s_i}, 0\}$. Then $x'$ an optimal solution of $I'$ is also an optimal solution of $I$.*

# 4 Complexity of min-max and min-max regret versions

Complexity of the min-max (regret) versions has been studied extensively during the last decade. Clearly, the min-max version of problem $\mathcal{P}$ is at least as hard as $\mathcal{P}$ since $\mathcal{P}$ is a particular case of DISCRETE MIN-MAX $\mathcal{P}$ when the scenario set contains only one scenario and $\mathcal{P}$ is a particular case of INTERVAL MIN-MAX $\mathcal{P}$ when all coefficients are degenerate. Similarly, the min-max regret version of $\mathcal{P}$ is at least as difficult as $\mathcal{P}$ since $\mathcal{P}$ reduces to DISCRETE (INTERVAL) MIN-MAX REGRET $\mathcal{P}$ when the scenario set is reduced to one scenario. For this reason, in the following, we consider only min-max (regret) versions of polynomial or pseudo-polynomial time solvable problems, for which we could hope to preserve the complexity.

For the discrete scenario case, Kouvelis and Yu [30] give complexity results for several combinatorial optimization problems, including SHORTEST PATH, SPANNING TREE, ASSIGNMENT, and KNAPSACK. In general, these versions are shown to be harder than the classical versions. More precisely, if the number of scenarios is non-constant (the number of scenarios is part of the input), these problems become strongly *NP*-hard, even when the classical problems are solvable in polynomial time. On the other hand, for a constant number of scenarios, it is only partially known if these problems are strongly or weakly *NP*-hard. Indeed, the reductions described in [30] to prove the *NP*-hardness are based on transformations from the partition problem which is known to be weakly *NP*-hard [23]. These reductions give no indications on the precise status of these problems. Most of the problems which are known to be weakly *NP*-hard are those for which there exists a pseudo-polynomial algorithm based on dynamic programming [30] (MIN-MAX (REGRET) SHORTEST PATH, MAX-MIN KNAPSACK, MIN-MAX REGRET KNAPSACK, MIN-MAX (REGRET) SPANNING TREE on grid graphs, ...)

As an example for this family of algorithms, we describe a pseudo-polynomial algorithm to solve DISCRETE MAX-MIN KNAPSACK. The standard procedure to solve the classical knapsack problem is based on dynamic programming. We give an extension to the multi-scenario case which is very similar to the procedure presented in [22] by Erlebach, Kellerer and Pferschy to solve multi-objective knapsack problem. This procedure is easier and more efficient than the one originally presented by Yu [49].

Consider a knapsack instance where each item $i$ has a weight $w_i$ and profit $p_i^s$, for $i = 1, \dots, n$ and $s = 1, \dots, k$, and a capacity $b$.

Let $W_i(v_1, \ldots, v_k)$ denote the minimum weight of any subset of items among the first $i$ items with profit $v_s$ in each scenario $s \in S$. The initial condition of the algorithm is given by setting $W_0(0, \ldots, 0) = 0$ and $W_0(v_1, \ldots, v_k) = b+1$ for all other combinations. The recursive relation is given by:

If $v_s \geq p_i^s$ for all $s \in S$, then

$$W_i(v_1, \ldots, v_k) = \min\{W_{i-1}(v_1, \ldots, v_k), W_{i-1}(v_1 - p_i^1, \ldots, v_k - p_i^k) + w_i\}$$

else $$W_i(v_1, \ldots, v_k) = W_{i-1}(v_1, \ldots, v_k)$$

for $i = 1, \ldots, n$.

Each entry of $W_n$ satisfying $W_n(v_1, \ldots, v_k) \leq b$ corresponds to a feasible solution with profit $v_s$ in scenario $s$. The set of items leading to a feasible solution can be determined easily using standard bookkeeping techniques. The feasible solutions are collected and an optimal solution to DISCRETE MAX-MIN KNAPSACK is obtained by picking a feasible solution minimizing $\max_{s\in S} v_s$. Since, $v_s \in \{0, \ldots, \sum_{i=1}^n p_i^s\}$, the running time of this approach is given by going through the complete profit space for every item that is $O(n(\max_{s\in S} \sum_{i=1}^n p_i^s)^k)$. The algorithm presented by Yu [49] has a running time $O(nb(\max_{s\in S} \sum_{i=1}^n p_i^s)^k)$.

SPANNING TREE is not known to have a dynamic programming scheme but can be solved, for instance, using Kruskal's algorithm or Prim's algorithm. A specific pseudo-polynomial algorithm is given in [3] for DISCRETE MIN-MAX SPANNING TREE using an extension of the matrix tree theorem to the multi-scenario case.

Aissi, Bazgan, and Vanderpooten investigate in [2] the complexity of min-max and min-max regret versions of two closely related polynomial-time solvable problems: CUT and $s{-}t$ CUT. For a constant number of scenarios, the complexity status of these problems is widely contrasted. More precisely, DISCRETE MIN-MAX (REGRET) CUT are polynomial, whereas DISCRETE MIN-MAX (REGRET) $s - t$ CUT are strongly *NP*-hard even for two scenarios. It is interesting to observe that $s - t$ CUT is the first polynomial-time solvable problem, whose min-max and min-max regret versions are strongly *NP*-hard.

Network location models are characterized by two types of parameters (weights of nodes and lengths of edges) and by a location site (on nodes or on edges). Kouvelis and Yu [30] show that DISCRETE MIN-MAX (REGRET) 1-MEDIAN can be solved in polynomial time for trees with scenarios both on node weights and edge lengths. If the location sites are limited to nodes, DISCRETE MIN-MAX (REGRET) 1-MEDIAN can be solved in polynomial time using an exhaustive search algorithm.

The complexity of the min-max (regret) versions of scheduling problems has also been investigated. Daniels and Kouvelis [18] show that DISCRETE MIN-MAX (REGRET) $1||\sum w_j C_j$ are *NP*-hard. Aloulou and Della Croce [6] show that DISCRETE MIN-MAX

$1|prec|f_{max}$ is polynomial-time solvable whereas DISCRETE MIN-MAX $1||\sum U_j$ is *NP*-hard.

In the interval scenario case, extensive research has been devoted for studying the complexity of min-max regret versions of various optimization problems including SHORTEST PATH [14], SPANNING TREE [8, 14], ASSIGNMENT [1], CUT and $s-t$ CUT [2]. In general, all these problems are shown to be strongly *NP*-hard.

For network locations problems, if edge lengths are uncertain/imprecise, INTERVAL MIN-MAX REGRET 1-MEDIAN is shown to be strongly *NP*-hard for general graphs [11] but polynomial-time solvable on trees [17]. However, if vertex weights are uncertain/imprecise, this problem is polynomial-time solvable [13]. For 1-CENTER, a closely related problem, INTERVAL MIN-MAX REGRET 1-CENTER is shown to be strongly *NP*-hard for general graphs if edge lengths are uncertain/imprecise but polynomial-time solvable on trees [13]. Nevertheless, if vertex weights are uncertain/imprecise, this problem is polynomial-time solvable [12].

For scheduling problems, Kasperski [27] shows that INTERVAL MIN-MAX REGRET $1|prec|f_{max}$ is polynomial-time solvable. Lebedev and Averbakh [31] show that INTERVAL MIN-MAX REGRET $1||\sum w_jC_j$ is *NP*-hard.

Tables 3-5 summarize the complexity of min-max (regret) versions of several classical combinatorial optimization problems. For all studied problems, min-max and min-max regret versions have the same complexity in the discrete scenario case. In general, there is no reduction between the min-max and the min-max regret versions or vice versa. However, in all known reductions establishing the *NP*-hardness of min-max versions [30, 1, 2], the obtained instances of min-max versions have been designed so as to get an optimal value on each scenario equal to zero. This way, these reductions also imply the *NP*-hardness of the min-max regret associated versions. Comparing now the complexity of the min-max regret version in the discrete scenario case and in the interval scenario case, we observe some slight differences. As shown by Averbakh [10], there even exists a simple combinatorial optimization problem, the selection problem, whose min-max regret version is solvable in polynomial time in the interval scenario case whereas it is *NP*-hard even for two scenarios.

Table 3: Complexity of the min-max (regret) versions of classical combinatorial problems (constant number of scenarios)

| Problems | Constant | |
|---|---|---|
| | Min-max | Min-max regret |
| SHORTEST PATH | *NP*-hard, pseudo-poly [30] | *NP*-hard, pseudo-poly [30] |
| SPANNING TREE | *NP*-hard [30], pseudo-poly [3] | *NP*-hard [30], pseudo-poly [3] |
| ASSIGNMENT | *NP*-hard [30] | *NP*-hard [30] |
| KNAPSACK | *NP*-hard, pseudo-poly [30] | *NP*-hard, pseudo-poly [30] |
| CUT | polynomial [7] | polynomial [2] |
| $s-t$ CUT | strongly *NP*-hard [2] | strongly *NP*-hard [2] |

Table 4: Complexity of the min-max (regret) versions of classical combinatorial problems (non-constant number of scenarios)

| Problems | Non-constant | |
|---|---|---|
| | Min-max | Min-max regret |
| SHORTEST PATH | strongly *NP*-hard [30] | strongly *NP*-hard [30] |
| SPANNING TREE | strongly *NP*-hard [30] | strongly *NP*-hard [4] |
| ASSIGNMENT | strongly *NP*-hard [1] | strongly *NP*-hard [1] |
| KNAPSACK | strongly *NP*-hard [30] | strongly *NP*-hard [4] |
| CUT | strongly *NP*-hard [2] | strongly *NP*-hard [2] |
| $s-t$ CUT | strongly *NP*-hard [2] | strongly *NP*-hard [2] |

# 5 Approximation of the min-max (regret) versions

The approximation of the min-max (regret) versions of classical combinatorial optimization problems has received great attention recently. Aissi, Bazgan, and Vanderpooten initiated in [4] a systematic study of this question in the discrete scenario case. More precisely, they investigate the relationships between min-max, min-max regret and multi-objective versions, and show the existence, in the case of a constant number of scenarios, of fully polynomial-time approximation schemes (fptas) for min-max (regret) versions of several classical optimization problems. They also study the case of a non-constant number of scenarios. In this setting, they provide non-approximability results for min-max (regret) versions of shortest path and spanning tree. In [3] they adopt an alternative

Table 5: Complexity of min-max regret versions of classical combinatorial problems (interval scenario case)

| Problems | Min-max regret |
|---|---|
| SHORTEST PATH | strongly *NP*-hard [14] |
| SPANNING TREE | strongly *NP*-hard [14] |
| ASSIGNMENT | strongly *NP*-hard [1] |
| KNAPSACK | *NP*-hard |
| CUT | polynomial [2] |
| $s-t$ CUT | strongly *NP*-hard [2] |

perspective and develop a general approximation scheme, using the scaling technique, which can be applied to min-max (regret) versions of some problems, provided that some conditions are satisfied. The advantage of this second approach is that the resulting fptas usually have much better running times than those derived using multi-objective fptas for the multi-objective versions.

The interval scenario case has been much less studied. We only report one general result due to Kasperski and Zieliński [28].

## 5.1 Discrete scenario case

As a first general result, Proposition 1 gives a $k$-approximation algorithm for DISCRETE MIN-MAX (REGRET) $\mathcal{P}$ when the underlying problem $\mathcal{P}$ is polynomial-time solvable. In some cases, however, we can derive fptas for some problems.

### 5.1.1 Relationships between min-max (regret) and multi-objective versions

Exploiting the results mentioned in section 3.1.2 concerning the relationships with the multi-objective version, we can derive approximation algorithms, with a noticeable difference between the min-max and the min-max regret versions.

**Min-max version**

**Proposition 11 ([4])** *Given a minimization problem $\mathcal{P}$, for any function $f : \mathbb{N} \rightarrow (1, \infty)$, if* MULTI-OBJECTIVE $\mathcal{P}$ *has a polynomial-time $f(n)$-approximation algorithm, then* DISCRETE MIN-MAX $\mathcal{P}$ *has a polynomial-time $f(n)$-approximation algorithm.*

**Proof :** Let $F$ be an $f(n)$-approximation of the set of efficient solutions. Since at least one optimal solution $x^*$ for DISCRETE MIN-MAX $\mathcal{P}$ is efficient as shown in Proposition 5, there exists a solution $y \in F$ such that $val(y,s) \leq f(n)val(x^*,s)$, for all $s \in S$. Consider among set $F$ a solution $z$ that has a minimum $\max_{s \in S} val(z,s)$. Thus, $\max_{s \in S} val(z,s) \leq \max_{s \in S} val(y,s) \leq f(n) \max_{s \in S} val(x^*,s)$. □

**Corollary 2 ([4])** *For a constant number of scenarios,* DISCRETE MIN-MAX SHORTEST PATH, DISCRETE MIN-MAX SPANNING TREE, *and* DISCRETE MAX-MIN KNAPSACK *have an fptas.*

**Proof :** For a constant number of objectives, multi-objective versions of SHORTEST PATH, SPANNING TREE, and KNAPSACK have an fptas as shown by Papadimitriou and Yannakakis in [40]. □

**Min-max regret version**

As shown in Proposition 6, at least one optimal solution $x^*$ for DISCRETE MIN-MAX REGRET $\mathcal{P}$ is efficient. Unfortunately, given $F$ an $f(n)$-approximation of the set of efficient solutions, a solution $x \in F$ with a minimum $R_{max}(x)$ is not necessarily an $f(n)$-approximation for the optimum value since the minimum maximum regret could be very small compared with the error that was allowed in $F$.

However, it is sometimes possible to derive an approximation algorithm working directly in the regret space instead of the criterion space. This way, an fptas is obtained in [4] for DISCRETE MIN-MAX REGRET SHORTEST PATH using a dynamic programming algorithm that computes at each stage the set of efficient vectors of regrets.

### 5.1.2 A general approximation scheme in the discrete scenario case

A standard approach for designing fptas for *NP*-hard problems is to use the *scaling technique* [43] in order to transform an input instance into an instance that is easy to solve. Then an optimal solution for this new instance is computed using a pseudo-polynomial algorithm. This solution will serve as an approximate solution. This general scheme usually relies on the determination of a lower bound that is polynomially related to the maximum value of the original instance. However, for the min-max (regret) versions, we need to adapt this scheme in order to obtain an fptas.

**Proposition 12 ([3])** *Given a problem* DISCRETE MIN-MAX (REGRET) $\mathcal{P}$, *if*

1. *for any instance $I$, a lower and an upper bound $L$ and $U$ of the optimal value $opt$ can be computed in time $p(|I|)$, such that $U \leq q(|I|)L$, where $p$ and $q$ are two polynomials with $q$ non decreasing and $q(|I|) \geq 1$, and*

2. *there exists an algorithm that finds for any instance $I$ an optimal solution in time $r(|I|, U)$ where $r$ is a non decreasing polynomial,*

*then* DISCRETE MIN-MAX (REGRET) $\mathcal{P}$ *has an fptas.*

We discuss now the two conditions of the previous theorem. The first condition can be satisfied easily for polynomial-time solvable problems as shown in Proposition 1 where we exhibit bounds $L$ and $U$ such that $U \leq kL$. Furthermore, if $\mathcal{P}$ is polynomially approximable, then the first condition of Proposition 12 can also be satisfied for DISCRETE MIN-MAX $\mathcal{P}$ using Proposition 1. More precisely, if $\mathcal{P}$ is $f(n)$-approximable where $f(n)$ is a polynomial, given an instance $I$ of DISCRETE MIN-MAX $\mathcal{P}$, let $x'$ be an $f(|I|/k)$-approximate solution in $I'$ (defined as in the proof of Proposition 1), then we have $L = \frac{1}{f(|I|/k)} \sum_{s \in S} \frac{1}{k} val(x', s)$ and $U = \max_{s \in S} val(x', s)$, and thus $U \leq kf(|I|/k)L$.

The second condition of Proposition 12 can be weakened for DISCRETE MIN-MAX $\mathcal{P}$ by requiring only a pseudo-polynomial algorithm, that is an algorithm polynomial in $|I|$ and $max(I) = \max_{i,s} c_i^s$. Indeed, knowing an upper bound $U$, we can eliminate any variable $x_i$ such that $c_i^s > U$ on at least one scenario $s \in S$. Condition 2 is then satisfied by applying the pseudo-polynomial algorithm on this modified instance.

Min-max (regret) versions of some problems, like SHORTEST PATH, KNAPSACK, admit pseudo-polynomial time algorithms based on dynamic programming [30]. For some dynamic programming formulations, we can easily obtain algorithms satisfying condition 2, by discarding partial solutions with value more than $U$ on at least one scenario. For other problems, like SPANNING TREE, which are not known to admit pseudo-polynomial algorithms based on dynamic programming, specific algorithms are required [3].

**Corollary 3 ([3])** *For a constant number of scenarios,* DISCRETE MIN-MAX (REGRET) SHORTEST PATH, DISCRETE MIN-MAX (REGRET) SPANNING TREE *have an fptas.*

The resulting fptas obtained using Corollary 3 are much more efficient than those obtained using Corollary 2.

Table 6 summarizes approximation results in the discrete scenario case for several combinatorial optimization problems.

Table 6: Approximation of the min-max and min-max regret versions in the discrete scenario case

| Problems | Constant | | Non-constant | |
|---|---|---|---|---|
| | Min-max | Min-max regret | Min-max | Min-max regret |
| SHORTEST PATH | fptas [46] | fptas [4, 3] | not $(2-\varepsilon)$ apx. [4] | not $(2-\varepsilon)$ apx. [4] |
| SPANNING TREE | fptas [4, 3] | fptas [4, 3] | not $(\frac{3}{2}-\varepsilon)$ apx. [4] | not $(\frac{3}{2}-\varepsilon)$ apx. [4] |
| KNAPSACK | fptas [4] | non apx. [4] | non apx. [4] | non apx. [4] |

## 5.2 Interval scenario case

In this case, most of the min-max regret versions of classical combinatorial optimization problems become strongly *NP*-hard (see Table 5). Kasperski and Zieliński [28] give a 2-approximation algorithm for the min-max regret versions of polynomial-time solvable problems as a direct consequence of Proposition 9.

# 6 Resolution of the min-max (regret) versions

We review in this section exact procedures for solving the min-max and min-max regret versions of classical combinatorial optimization problems.

## 6.1 Discrete scenario case

Kouvelis and Yu [30] implement a branch-and-bound procedure to solve the min-max (regret) versions of several combinatorial optimization problems. A lower bound on each node of the arborescence is calculated using surrogate relaxation as explained in the following.

Consider DISCRETE MIN-MAX REGRET $\mathcal{P}$ which can be stated as:

$$\begin{cases} \min y \\ y \geq val(x,s) - val^*_s, \ \forall s \in S \\ x \in X, y \geq 0 \end{cases} \tag{3}$$

Let $\mu = (\mu_1, \ldots, \mu_k)$ be a multiplier vector such that $\mu_s \geq 0$, $\forall s \in S$, and $\sum_{s \in S} \mu_s = 1$. The surrogate relaxation of (3) is formulated as follows:

$$\begin{cases} L(\mu) = \min \sum_{s \in S} \mu_s (val(x,s) - val^*_s) \\ x \in X \end{cases} \tag{4}$$

The efficiency of the relaxation procedure rests on finding a vector $\mu^*$ providing the best lower bound $L(\mu^*) \leq opt$ where $opt$ is the optimum value of the input considered. A sub-gradient procedure is used in [30] to find the tightest one. The later is based on solving, at each iteration, problem (4) with a multiplier vector $\mu = (\mu_1, \ldots, \mu_k)$ and the result is exploited in order to find a new multiplier vector that improves the bound given by the surrogate relaxation. Furthermore, the optimal solution $x(\mu)$ of (4) is a feasible solution of (3) and thus provides an upper bound $U(\mu) = \max_{s \in S}(val(x(\mu), s) - val_s^*)$ on the optimal value. The best upper bound obtained so far during the execution of the sub-gradient procedure is saved.

The Best-Node First search strategy is adopted in the branch-and-bound procedure presented in [30]. The choice of the branching variable is guided by the surrogate relaxation.

DISCRETE MIN-MAX $\mathcal{P}$ can be solved in a similar way by removing $val_s^*$ in (3) and (4).

## 6.2 Interval scenario case

In the interval scenario case, Inuiguchi and Sakawa [24] study the min-max regret version of linear programs. Their algorithm is based on a relaxation procedure developed initially by Shimizu and Aiyochi [44]. Karasan, Pinar and Yaman [26, 48] formulate the min-max regret versions of SHORTEST PATH and SPANNING TREE as mixed integer linear programs. A generalization of this formulation to problems with zero duality gap is presented in section 6.2.1. Montemanni and Gambardella [36, 37] propose branch-and-bound algorithms for solving these problems. Recently, Aissi, Vanderpooten and Vanpeperstraete [5], Montemanni [34], Montemanni and Gambardella [38], and Montemanni, Barta and Gambardella [35] proposed a relaxation procedure for solving the min-max regret versions of problems from $\mathcal{C}$.

In this section, we give a formulation of the min-max regret versions of problems with a zero duality gap (section 6.2.1) and we present a general relaxation procedure to solve the min-max regret versions of polynomial-time solvable or *NP*-hard problems (section 6.2.2).

### 6.2.1 Formulation of the min-max regret versions of problems with a zero duality gap

Consider a minimization problem $\mathcal{P} \in \mathcal{C}$ with a zero duality gap and where the feasible solution set is defined by $X = \{x : Ax \geq b,\ x \in \{0,1\}^n\}$. The maximum regret

$R_{max}(x)$ of a solution $x$ is given as follows:

$$R_{max}(x) = \max_{s \in S, y \in X} \{c^s x - c^s y\} = c^-(x)x - c^-(x)x^*_{c^-(x)} = \overline{c}x - \min_{z \in X} c^-(x)z$$

where $\overline{c} = (\overline{c}_1, \ldots, \overline{c}_n)$ and $x^*_s$ is an optimal solution for scenario $s$.

Therefore, INTERVAL MIN-MAX REGRET $\mathcal{P}$ can be written as follows:

$$\min_{x \in X} R_{max}(x) = \min_{x \in X} \{\overline{c}x - \min_{z \in X} c^-(x)z\} \tag{5}$$

Since $\mathcal{P}$ has a zero duality gap, we can replace in (5) the problem $\min_{z \in X} c^-(x)z$ by its dual:

$$\max_{y \in Y} b^t y$$

where $Y = \{y : A^t y \leq c^-(x),\ y \geq 0\}$. Consequently, INTERVAL MIN-MAX REGRET $\mathcal{P}$ can be formulated as follows:

$$\min_{x \in X,\ y \in Y} \{\overline{c}x - b^t y\}$$

In the following, we illustrate this approach with the assignment problem. A linear programming formulation of this problem is given as follows:

$$\begin{cases} \min \sum_{i,j=1}^n c_{ij} x_{ij} \\ \text{s.t.} \\ \sum_{j=1}^n x_{ij} = 1;\ i = 1, \ldots, n \\ \sum_{i=1}^n x_{ij} = 1;\ j = 1, \ldots, n \\ x_{ij} \geq 0;\ i, j = 1, \ldots, n \end{cases} \tag{6}$$

The dual of (6) is given by

$$\begin{cases} \max \sum_{i=1}^n (u_i + v_j) \\ \text{s.t.} \\ u_i + v_j \leq c_{ij};\ \ i, j = 1, \ldots, n \\ u_i, v_j \gtreqless 0;\ i, j = 1, \ldots, n \end{cases}$$

Thus, a formulation of INTERVAL MIN-MAX REGRET ASSIGNMENT is

$$\begin{cases} \min \sum_{i,j=1}^n \overline{c}_{ij} x_{ij} - \sum_{i=1}^n (u_i + v_j) \\ \text{s.t.} \\ \sum_{j=1}^n x_{ij} = 1;\ i = 1, \ldots, n \\ \sum_{i=1}^n x_{ij} = 1;\ j = 1, \ldots, n \\ u_i + v_j \leq \underline{c}_{ij} + (\overline{c}_{ij} - \underline{c}_{ij}) x_{ij};\ \ i, j = 1, \ldots, n \\ x_{ij} \in \{0, 1\};\ i, j = 1, \ldots, n \\ u_i, v_j \gtreqless 0;\ i, j = 1, \ldots, n \end{cases}$$

### 6.2.2 A relaxation procedure for the min-max regret versions

A general linear formulation of min-max regret versions is given by:

$$\textit{MMR}\begin{cases}\min r\\ \text{s.t.}\\ r \geq c^s x - c^s x^*_s, \quad \forall s \in S\\ x \in X, r \geq 0\end{cases}$$

where constraint $r \geq c^s x - c^s x^*_s$ is referred to as a cut of type 1.

From Proposition 7, we know that, in the interval scenario case, we can restrict the scenario set $S$ to the set of extreme scenarios. However, since the number of extreme scenarios is up to $2^n$, formulation *MMR* is exponential.

In order to handle this difficulty, a standard approach is to use a relaxation procedure which starts with an initial subset of scenarios $S^0$ and iteratively takes into account one additional scenario, by adding a cut of type 1, until we can establish that the current solution is optimal.

Considering the following formulation, defined at iteration $h$ with a current subset of scenarios $S^h$:

$$\textit{MMR}^h\begin{cases}\min r\\ \text{s.t.}\\ r \geq c^s x - c^s x^*_s, \quad \forall s \in S^h\\ x \in X, r \geq 0\end{cases}$$

we denote by $r^h$ and $x^h$ the optimal value and an optimal solution of *MMR*$^h$ at step $h$.

The details of the relaxation procedure, originally proposed in [24, 32, 33] for solving min-max regret versions of linear programming problems, are given as follows:

**Algorithm 1** Relaxation procedure

1: Initialisation. $LB \leftarrow 0$, $S^0 \leftarrow \emptyset$, $h \leftarrow 0$, and construct $x^0 \in X$.
2: Compute $\widehat{c}^h$ and $R_{max}(x^h)$ by solving $\max_{s\in S, x\in X}\{c^s x^h - c^s x\}$.
3: **if** $R_{max}(x^h) \leq LB$ **then**
4: END: $x^h$ is an optimal solution.
5: $S^{h+1} \leftarrow S^h \cup \{\widehat{c}^h\}$.
6: $h \leftarrow h+1$, compute $r^h$ and $x^h$ by solving *MMR*$^h$, $LB \leftarrow r^h$ and go to step 2.

In the linear programming case, step 2 is the most computationally demanding part of the algorithm since it requires solving a quadratic programming problem. On the other hand, step 6 requires only solving a linear programming problem.

This algorithm can be extended to 0-1 linear programming problems. In this case, we do not need to solve a quadratic programming problem at step 2 since we know, from Proposition 7, that we can take as scenario $\widehat{c}^h$ the worst case scenario for for $x^h$, $c^-(x^h)$. Hence, step 2 of Algorithm 1 can be replaced by:

2: $\widehat{c}^h \leftarrow c^-(x^h)$, compute an optimal solution $x^*_{\widehat{c}^h}$ of $\min_{x \in X} \widehat{c}^h x$,
$R_{max}(x^h) \leftarrow \widehat{c}^h x^h - \widehat{c}^h x^*_{\widehat{c}^h}$.

Now step 6 becomes computationally expensive since *MMR*$^h$ is a mixed integer programming problem whose difficulty increases with the number of added cuts. This explains why a straightforward application of this algorithm to solve min-max regret versions of 0-1 linear programming problems is not satisfactory. We can improve significantly the performance of the algorithm by using alternative cuts, called cuts of type 2, which were found independently in [5] and [34, 38]. These cuts are defined as follows:

$$r \geq \overline{c}x - c^-(x)x^*_{c^-(x^h)}$$

These cuts can be derived easily for a 0-1 linear programming problem, since $R_{max}(x) \geq \overline{c}x - c^-(x)y$ for any $x, y \in X$, and in particular for $y = x^*_{c^-(x^h)}$.

**Proposition 13** *Given a minimization problem $\mathcal{P} \in \mathcal{C}$, cuts of type 2 dominate cuts of type 1.*

**Proof :** We need to prove that for any $x, x^h \in X$, we have $\overline{c}x - c^-(x)x^*_{c^-(x^h)} \geq c^-(x^h)x - c^-(x^h)x^*_{c^-(x^h)}$. Thus, we consider the following difference:

$$\overline{c}x - c^-(x)x^*_{c^-(x^h)} - c^-(x^h)x + c^-(x^h)x^*_{c^-(x^h)} = \sum_{i=1}^{m}(\overline{c}_i - \underline{c}_i)(x_i - x_i x^*_{c^-(x^h),i} - x_i x^h_i + x^h_i x^*_{c^-(x^h),i})$$

If $x^h_i = 1$, we have $(\overline{c}_i - \underline{c}_i)(x^*_{c^-(x^h),i} - x_i x^*_{c^-(x^h),i}) \geq 0$ for $i = 1, \ldots, n$. If $x^h_i = 0$, we have $(\overline{c}_i - \underline{c}_i)(x_i - x_i x^*_{c^-(x^h),i}) \geq 0$ for $i = 1, \ldots, n$. □

# 7 Conclusions

We reviewed in this paper motivations, complexity, approximation, and exact resolution for the min-max and min-max regret versions of several combinatorial optimization problems. We conclude this survey by listing some open questions.

For all studied problems, min-max and min-max regret versions have the same complexity in the discrete scenario case. As observed at the end of section 4, there is no general reduction between the min-max and the min-max regret versions or vice versa, except for bottleneck problems. The construction of such a reduction would be interesting since only one algorithm (or *NP*-hardness proof) would be sufficient for both versions.

Almost all problems discussed in this paper have a clear complexity status, see, e.g., Tables 3-5. However, a few *NP*-hard problems still need to be precisely classified (weakly or strongly *NP*-hard). As a first example, INTERVAL MIN-MAX REGRET KNAPSACK is clearly *NP*-hard, but its precise status is not known. A second example is DISCRETE MIN-MAX (REGRET) ASSIGNMENT for a constant number of scenarios. The original proofs of the *NP*-hardness of these problems are obtained in [30] using a reduction from partition, which has a pseudo-polynomial time algorithm [23]. Therefore, it remains an open question whether these problems have pseudo-polynomial time algorithms or are strongly *NP*-hard.

Approximation results, which are quite recent, should be studied more thoroughly. In particular, for a non-constant number of scenarios, there is a gap between the general $k$-approximation result given at the beginning of section 5.1 and specific negative results indicated in Table 6. In the interval scenario case, finding better approximation algorithms than the general 2-approximation algorithm referred in section 5.2, for the min-max regret versions of specific polynomial-time solvable problems, is also an interesting research perspective.

Finally, the efficient exact resolution of the min-max and min-max regret versions of combinatorial optimization problems remains a computationally challenging question.

# References

[1] H. Aissi, C. Bazgan, and D. Vanderpooten. Complexity of the min-max and min-max regret assignment problem. *Operations Research Letters*, 33(6):634–640, 2005.

[2] H. Aissi, C. Bazgan, and D. Vanderpooten. Min-max (regret) versions of cut problems. In *Proceedings of the 16th International Symposium on Algorithms and Computation (ISAAC 2005), Sanya, Hainan, China*, LNCS 3827, pages 789–798. Springer-Verlag, 2005.

[3] H. Aissi, C. Bazgan, and D. Vanderpooten. Approximating min-max (regret) versions of some polynomial problems. In *Proceedings of the 12th Annual International Computing and Combinatorics Conference (COCOON 2006), Taipei, Taiwan*, LNCS 4112, pages 428–438. Springer-Verlag, 2006.

[4] H. Aissi, C. Bazgan, and D. Vanderpooten. Approximation of min-max and min-max regret versions of some combinatorial optimization problems. *European Journal of Operational Research*, 179(2):281–290, 2007.

[5] H. Aissi, D. Vanderpooten, and J.M. Vanpeperstraete. Robust approaches for the data association problem. In *Proceedings of the 8th International Conference on Information Fusion (FUSION 2005)*, Philadelphia, USA, 2005.

[6] M.A. Aloulou and F. Della Croce. On the complexity of single machine scheduling problems under scenario-based uncertainty. Annales du LAMSADE N˚4-5, pages 1–14, 2005.

[7] A. Armon and U. Zwick. Multicriteria global minimum cuts. In *Proceedings of the 15th International Symposium on Algorithms and Complexity (ISAAC 2004), Hong Kong, China*, LNCS 3341, pages 65–76. Springer-Verlag, 2004.

[8] I. D. Aron and P. Van Hentenryck. On the complexity of the robust spanning tree with interval data. *Operations Research Letters*, 32(1):36–40, 2004.

[9] I. Averbakh. Minmax regret solutions for minimax optimization problems with uncertainty. *Operations Research Letters*, 27:57–65, 2000.

[10] I. Averbakh. On the complexity of a class of combinatorial optimization problems with uncertainty. *Mathematical Programming*, 90:263–272, 2001.

[11] I. Averbakh. Complexity of robust single facilitylocation problems on networks with uncertain edge lengths. *Discrete Applied Mathematics*, 127:505–522, 2003.

[12] I. Averbakh and O. Berman. Minimax regret p-center location on a network with demand uncertainty. *Location Science*, 5(4):247–254, 1997.

[13] I. Averbakh and O. Berman. An improved algorithm for the min-max regret median problem on a tree. *Networks*, 41(2):97–103, 2003.

[14] I. Averbakh and V. Lebedev. Interval data min-max regret network optimization problems. *Discrete Applied Mathematics*, 138(3):289–301, 2004.

[15] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming series B*, 98:49–71, 2003.

[16] R. D. Carr, H. J. Greenberg, W. E. Hart, G. Konjevod E. Lauer, H. Lin, T. Morrison, and C. A. Phillips. Robust optimization of contaminant sensor placement for community water systems. *Mathematical Programming, Serie B*, 107(1–2):337–356, 2006.

[17] B. Chen and C. S. Lin. Min-max regret robust 1-median location on a tree. *Networks*, 31:93–103, 1998.

[18] R. L. Daniels and P. Kouvelis. Robust scheduling to hedge against processing time uncertainty in single-stage production. *Management Science*, 41(2):363–376, 1995.

[19] M.S. Daskin, S.M. Hesse, and C.S. ReVelle. $\alpha$-reliable p-minimax regret: a new model for strategic facility location modeling. *Location Science*, 5:227–246, 1997.

[20] A. Dodonova and Y. Khoroshilov. Applications of regret theory to asset pricing. Technical report, University of Ottawa, 2005.

[21] M. Ehrgott. *Multicriteria Optimization*. Springer, 2nd edition, 2005.

[22] T. Erlebach, H. Kellerer, and U. Pferschy. Approximating multiobjective knapsack problems. *Management Science*, 48(12):1603–1612, 2002.

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

[24] M. Inuiguchi and M. Sakawa. Minimax regret solution to linear programming problems with interval objective function. *European Journal of Operational Research*, 86(3):526–536, 1995.

[25] R. Kalai, M.A. Aloulou, P. Vallin, and D. Vanderpooten. Robust 1-median location problem on a tree. In *Proceedings of Euro Conference for Young OR Researchers and Practitioners (ORP3 2005), Valencia, Spain*, 2005.

[26] O. E. Karaşan, M. C. Pinar, and H. Yaman. The robust shortest path problem with interval data. 2001. Available from http://www.optimization-online.org.

[27] A. Kasperski. Minimizing maximal regret in single machine sequencing problem with maximum lateness criterion. *Operations Research Letters*, 33(4):431–436, 2005.

[28] A. Kasperski and P. Zieliński. An approximation algorithm for interval data min-max regret combinatorial optimization problems. *Information Processing Letters*, 97(5):177–180, 2006.

[29] A. O. Kazakçi, S. Rozakis, and D. Vanderpooten. Energy crop supply in france: a min-max regret approach. *Journal of the Operational Research Society*. To appear, 2007.

[30] P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*. Kluwer Academic Publishers, Boston, 1997.

[31] V. Lebedev and I. Averbakh. Complexity of minimizing the total flow time with interval data and minmax regret criterion. *Discrete Applied Mathematics*, 154:2167–2177, 2006.

[32] M. E. Mausser and M. Laguna. A heuristic to minimax absolute regret for linear programs with interval objective function coefficients. *European Journal of Operational Research*, 117(1):157–174, 1998.

[33] M. E. Mausser and M. Laguna. A new mixed integer formulation for the maximum regret problem. *International Transaction on Operations Research*, 5(5):398–403, 1998.

[34] R. Montemanni. A Benders decomposition approach for the robust spanning tree with interval data. *European Journal of Operational Research*, 174(3):1479–1490, 2006.

[35] R. Montemanni, J. Barta, and L. Gambardella. An exact algorithm for the robust travelling salesman problem with interval data. In *Proceedings of ODYSSEUS 2006*, pages 256–258, 2006.

[36] R. Montemanni and L. Gambardella. A branch and bound algorithm for the robust shortest path problem with interval data. *Operations Research Letters*, 32(3):225–232, 2005.

[37] R. Montemanni and L. Gambardella. A branch and bound algorithm for the robust spanning tree problem with interval data. *European Journal of Operational Research*, 161(3):771–779, 2005.

[38] R. Montemanni and L. Gambardella. The robust shortest path problem with interval data via Benders decomposition. *4OR*, 3(4):315–328, 2005.

[39] J. M. Mulvey, R. J. Vanderbei, and S. A. Zenios. Robust optimization of large-scale systems. *Operations Research*, 43(2):264–281, 1995.

[40] C. H. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *IEEE Symposium on Foundations of Computer Science (FOCS 2000), Redondo Beach, California, USA*, pages 86–92, 2000.

[41] J. Rosenhead, M. Elton, and S.K. Gupta. Robustness and optimality as criteria for strategic decisions. *Operational Research Quarterly*, 23(4):413–431, 1972.

[42] B. Roy. A missing link in OR-DA: robustness analysis. *Foundations of Computing and Decision Sciences*, 23(3):141–160, 1998.

[43] S. Sahni. General techniques for combinatorial approximation. *Operations Research*, 25(6):920–936, 1977.

[44] K. Shimizu and E. Aiyoshi. Necessary conditions for min-max problems and algorithms by a relaxation procedure. *IEEE Transactions on Automatic Control*, 25(1):62–66, 1980.

[45] P. Vincke. Robust solutions and methods in decision-aid. *Journal of Multicriteria Decision Analysis*, 8:181–187, 1999.

[46] A. Warburton. Approximation of Pareto optima in multiple-objective, shortest-path problems. *Operations Research*, 35(1):70–79, 1987.

[47] J. P. Watson, W. E. Hart, and R. Murray. Formulation and optimization of robust sensor placement problems for contaminant warning systems. Technical report, 2006.

[48] H. Yaman, O. E. Karaşan, and M. C. Pinar. The robust spanning tree problem with interval data. *Operations Research Letters*, 29(1):31–40, 2001.

[49] G. Yu. On the max-min 0-1 knapsack problem with robust optimization applications. *Operations Research*, 44(2):407–415, 1996.

# Flexible solutions in disjunctive scheduling : general formulation and study of the flow-shop case

Mohamed Ali Aloulou*, Christian Artigues†

**Résumé**

On considère le problème de pilotage d'atelier en temps réel et on propose de construire d'une façon procative une solution présentant de la flexibilité séquentielle pouvant être exploitée en temps réel pour absorber les incertitudes du modèle et les perturbations. Ceci peut être réalisé en définissant un ordre partiel des opérations au niveau de chaque machine, laissant la possibilité au décideur de compléter les décisions de séquencement en fonction de ses préférences et de l'état de l'exécution. Proposer une solution partielle n'est pertinent que si on est capable de donner une évaluation de tous les ordonnancements qui peuvent être obtenus par extension de cette solution. L'article répond alors aux deux questions suivantes : Quelle est la meilleure et la pire performance des ordonnancements pouvant être atteints par une règle de pilotage suivi par le décideur ? Comment construire une solution flexible avec des garanties de performance imposées ?

**Mots-clefs :** Ordonnancement disjonctif, flexibilité, évaluation de la performance dans le pire cas

**Abstract**

The purpose of this work is to provide the decision-maker a characterization of possible modifications of predictive schedules while preserving optimality. In the context of machine scheduling, the anticipated modifications are changes in the predictive order of operations on the machines. To achieve this goal, a flexible solution is provided. It represents a set of semi-active schedules and is characterized by a partial order on each machine, so that the total order can be set on-line, as required

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

†LAAS-CNRS, 7 avenue du Colonel Roche, 31077 Toulouse, France. artigues@laas.fr

by the decision maker. A flexible solution is optimal if all the complete schedules that can be obtained by extension are also optimal. We consider the problem of evaluating the worst case performance of flexible solutions in disjunctive scheduling. We show that this problem can be modeled as an elementary longest path problem in the disjunctive graph representing the scheduling problem with additional constraints. In the flow-shop context, we give a polynomial algorithm to solve the problem and propose a method to issue optimal flexible solutions. Computational experiments show that significant flexibility is obtained.

**Key words :** Disjunctive scheduling, flexibility, worst-case performance evaluation

# 1 Introduction

We consider a general non preemptive disjunctive problem in which a set of operations has to be scheduled on a set of machines, each operation requiring a fixed single machine during its execution and each machine being able to process only one operation simultaneously. The operations are linked by simple precedence constraints that do not necessarily form chains. Such a model encompasses the standard flow-shop and job-shop models.

An important issue in scheduling concerns the support provided to the end-user(s) for on-line schedule execution after the off-line scheduling phase, which consists in providing an optimal or suboptimal schedule.

In disjunctive scheduling, as soon as a regular minmax objective function is considered, the support for on-line scheduling often lies in providing for each machine the mandatory sequence of operations, and for each operation an earliest and a latest start time yielding operation slacks. The sequences and the time windows are such that scheduling the operations in the predetermined order and inside their time windows is feasible and keeps the objective function in a range of acceptable values. Such a flexibility provided to the end-user is referred to as temporal flexibility. This paper addresses the problem of providing more flexibility than the classical temporal one in disjunctive scheduling problems. As already considered in previous studies [2, 3, 6, 7, 11, 12, 27], this can be achieved by defining only a partial order of the operations on each machine, leaving to the end-user the possibility to make the remaining sequencing decisions. This is the principle of the groups of permutable operations model that has been studied by several authors [3, 6, 11, 12]. It is also called ordered assignment model by [27]. The group model sets restrictions on the proposed partial orders that we relax in this paper. Indeed we represent the partial orders through general precedence constraints between operations of the same machine, that have not to be distinguished from the structural precedence constraints.

A flexible solution allows to postpone some sequencing decisions and to hedge against some small to medium disruptions (raw material unavailability, small breakdowns, ...) with a minimum effort of computations [2, 15, 27]. It also permits to take into account some decision maker preferences that cannot be simply modeled or that may render the problem difficult to solve. For example, if two jobs can be executed in any order, without any influence on the performance, then the decision maker may prefer to sequence fist the job that can be processed rapidly to avoid the starvation of the next machine, or the job with a maximum number of successors to maintain enough flexibility in the future. He may also favor a job belonging to a privileged client or a job that has to be sent to a downstream shop for further processing.

Providing a partial solution through partial orders is useful in practice only if it can be assorted with an evaluation of the complete solutions that can be obtained by extension. More precisely, given a reasonable decision policy followed by the decision maker, the following questions have to be answered. Do there remain decisions (following the decision policy) leading to a feasible schedule ? What is the best and the worse objective function value reachable by the remaining set of decisions ? Answering these questions provides a performance guarantee if the given on-line policy is followed.

Two main issues are developed in this paper. The first concerns the evaluation of a flexible solution in the worst case. The second issue is the computation of optimal or near optimal flexible solutions. A flexible solution is optimal if all the complete schedules that can be obtained by extension are also optimal.

In the first part of the paper, we focus on the worst case performance evaluation of a flexible solution. We consider disjunctive scheduling problems with a minmax objective function. We assume that the decision maker follows a semi-active schedule policy to extend the proposed partial orders, which is relevant to any regular objective function. Recall that a schedule is called semi-active if the operations cannot be shifted to start earlier without changing the operation sequences or violating precedence constraints or release dates [4]. In this case, the worse objective function value can be determined by computing the worst-case completion time of each operation separately. We show that the problem of computing the worse completion time of an operation in all feasible semi-active schedules can be modeled as an elementary longest path problem in the disjunctive graph representing the scheduling problem with additional constraints. This statement provides a general framework for the majority of previous studies on representation and evaluation of set of solutions based on sufficient conditions in disjunctive problems [1, 3, 6, 7, 11, 12, 27].

In the special case of the flow-shop problem with release dates and additional precedence constraints, we give a polynomial algorithm that computes the maximal completion times of all operations in all feasible semi-active schedules. These results generalize some results previously established for the single machine version of this problem [1]. The proposed algorithm is used in a branch and bound method to compute feasible flexible solutions (w.r.t a common due-date) for a flow-shop problem denoted by $F|\tilde{d_i} = d|-$.

In Section 2, we present the worst-case computation problem in more details. We discuss the related work in Section 3. In Section 4, the longest path formulation of the problem is given. In Section 5, we present the polynomial algorithm for the flow-shop case. In Section 6, we present a method to compute feasible flexible solutions in the flow-shop context and provide computational experiments. Finally, we conclude in section 7.

# 2 Problem setting

We consider the following disjunctive scheduling problem. There is a set $N = \{1, \ldots, n\}$ of operations to be scheduled on $m$ machines. $m_i, p_i$ and $r_i$ denote the machine, processing time and release date of operation $i$, respectively. The release date is the earliest time when the operation processing can start. Each operation is associated with a non-decreasing cost function $f_i(C_i)$ of its completion time $C_i$. We introduce two dummy operations $0$ and $n+1$ such that $p_0 = p_{n+1} = 0$. This problem is represented by a disjunctive graph $\mathcal{G} = (V, C, D)$ [22]. $V$ is the set of vertices corresponding to operations $i \in N$ and the two dummy operations $0$ and $n+1$. $C$ is the set of conjunctive arcs representing the precedence constraints between the operations. Each conjunctive arc $(i, j)$ is valuated by $p_i$. $D$ is a set of pairs of disjunctive arcs $\{(i, j), (j, i)\}$ for each pair of operations $i, j \in N$ requiring the same machine for their execution ($m_i = m_j$). We have $D = \{\{(i, j), (j, i)\} | i \neq j$ and $m_i = m_j\}$. Arc $(i, j)$ represents the decision to sequence $i$ before $j$, whereas arc $(j, i)$ represents the decision to sequence $j$ before $i$ on the machine. In the remaining a pair of disjunctive arcs $\{(i, j), (j, i)\}$ is called a disjunction and denoted by $e_{ij}$ or $e_{ji}$.

Let $\mathcal{D}$ denote the set of all disjunctive arcs, i.e. $\mathcal{D} = \{(i, j) | e_{ij} \in D\}$. A selection $\pi$ is a (possibly empty) set of arcs such that $\pi \subseteq \mathcal{D}$ and $|\pi \cap e_{ij}| \leq 1$, for all $e_{ij} \in D$. Let $D(\pi) = \{e_{ij} \in D | e_{ij} \cap \pi = \emptyset\}$. A selection is complete if $D(\pi) = \emptyset$, otherwise it is partial. The disjunctive graph issued from a complete or partial selection $\pi$ is denoted by $\mathcal{G}(\pi) = (V, C \cup \pi, D(\pi))$. Given a set of arcs $E$, let $G(E)$ denote graph $(V, C \cup E)$. A complete selection $\pi$ is feasible if the graph $G(\pi) = (V, C \cup \pi)$ is acyclic. The completion time $C_i(\pi)$ of any operation $i \in N$ in the semi-active schedule derived from the complete feasible selection $\pi$ is equal to the length of the longest path in $G(\pi)$ from $0$ to $i$ plus $p_i$. Let $\Pi$ denote the set of feasible complete selections. The objective of the classical scheduling problem is to find a complete feasible selection $\pi \in \Pi$ such that a *regular* minmax objective function $F(C_1(\pi), \ldots, C_n(\pi)) = \max_{i=1,\ldots,n} f_i(C_i(\pi))$ is minimized.

Here, we assume the decision maker makes on-line the remaining sequencing decisions on each machine following a semi-active policy until obtaining a complete selection $\pi$. A feasible semi-active schedule can be obtained by a list scheduling algorithm as soon as $C$ is acyclic: sort the operations in a non decreasing order of their level in $G = (V, C)$ then sequence as soon as possible on its machine each operation according to this order. The first problem tackled in this paper is the following problem (SP) : Given a disjunctive graph $\mathcal{G} = (V, C, D)$, what is the worst case objective function value aver all feasible semi-active schedules, i.e compute $\max_{\pi \in \Pi} F(C_1(\pi), \dots, C_n(\pi))$ ?

To illustrate this problem, consider the following 2-machine and 4-job flow-shop problem. This gives 8 operations and the flow-shop context sets $m_1 = m_3 = m_5 = m_7 = 1$ and $m_2 = m_4 = m_6 = m_8 = 2$. Furthermore, we have $p_1 = 1$, $p_2 = 6$, $p_3 = 2$, $p_4 = 5$, $p_5 = 4$, $p_6 = 6$, $p_7 = 6$ and $p_8 = 1$. All release dates are equal to 0 except for $r_5 = 2$. All objective functions are the completion times of the activities. Let us consider additional precedence constraints $\{(1, 3), (1, 5), (1, 7), (3, 7), (2, 4), (2, 6), (6, 8)\}$. We obtain the disjunctive graph $\mathcal{G}$ displayed in Figure 1.

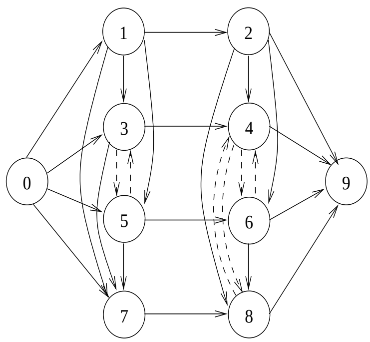



Figure 1: The disjunctive graph for a partial selection

The precedence constraints restrict the possible sequences to $(1, 3, 5, 7)$ and $(1, 5, 3, 7)$ on machine $1$ and $(2, 4, 6, 8)$, $(2, 6, 4, 8)$ and $(2, 6, 8, 4)$ on machine 2. We obtain the 6 schedules displayed in Figure 2. Note that such restriction cannot be modeled by the group of permutable operation representation used in [1, 6, 11, 12, 27]. The solution of problem SP is 20 which is the worst-case makespan value of the 6 semi-active schedules. Note that the optimal makespan of the flow-shop problem is 19.

The second problem considered in this paper is to propose a method allowing to compute optimal (or feasible w.r.t. a due date) flexible solutions with a maximum number of disjunctions $|D|$. Indeed, this quantity represents the amount of remaining decisions that can managed by the decision maker.

Consider the problem $F|d_i = d|-$ with $d = 20$ for the above example. The flexible solution represented in Figure 1 is feasible and represents the 6 schedules displayed in Figure 2. The number of disjunctions in this solution is $|D| = 3$.

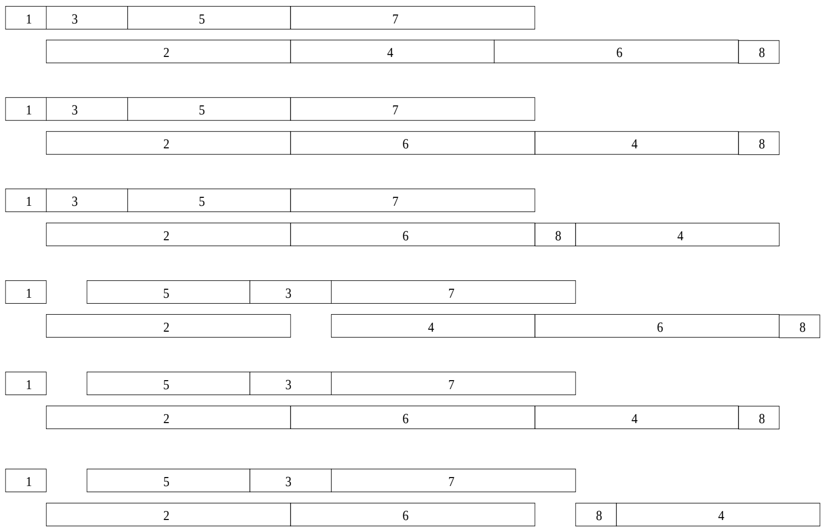



Figure 2: Sequences and semi-active schedules compatible with the partial selection

# 3 Related work

Problem SP can be viewed as a maximization problem of an objective function that is naturally minimized. Several works have aleady been proposed in this domain. In particular, Aloulou, Kovalyov and Portmann [1] adapt the traditional three-field notation $\alpha|\beta|\gamma$ to this class of problems.

They denote this family of considered maximization problems as $\alpha(sa)|\beta|(f \rightarrow \max)$. The first field $\alpha$ provides the shop environment. Here $\alpha \in \{1, F, J\}$ for respectively single machine (1), flow-shop ($F$) and job shop ($J$) problems. $sa$ indicates that we search for the worst schedule among all semi-active schedules, which correspond to the considered online scheduling policy. The second field gives additional constraints on operations. The third field contains information about the criterion to maximize.

To the best of our knowledge, the first related results we are aware are due to Posner [21]. Posner studied reducibility among single machine weighted completion time scheduling problems including minimization as well as maximization problems. In these problems, the jobs may have release dates and deadlines but there are no precedence constraints between the jobs. Besides, inserting idle times between the jobs is allowed.

Aloulou *et al* [1] studied several maximization versions in a single machine environment. They examined problems $1(sa)|\beta|(\gamma \rightarrow \max)$, where $\beta \subseteq \{r_i, prec\}$ and $\gamma \in \{f_{\max}, C_{\max}, L_{\max}, T_{\max}, \sum(w_i)C_i, \sum(w_i)U_i, \sum(w_i)T_i\}$. They showed that these problems are at least as easy as their minimization counterparts, except for problems $1(sa)||(\sum w_iT_i \rightarrow \max)$ and $1(sa)|r_i|(\sum w_iT_i \rightarrow \max)$, which are still open. In particular, problems $1(sa)|r_i, prec|(L_{\max} \rightarrow \max)$ and $1(sa)|r_i, prec|(T_{\max} \rightarrow \max)$ can be solved in $O(n^3)$ times while the minimization counterparts are strongly NP-hard, even if $prec = \emptyset$ [18].

This work is closely related to the work of the present paper. Indeed, problem (SP) can be denoted, in the Aloulou *et al* notation, as $\alpha(sa)|r_i, prec|(f_{\max} \rightarrow \max)$. Besides, we propose in section 5 an algorithm solving the flow-shop problem $F(sa)|r_i, prec_k|(f_{\max} \rightarrow \max)$, generalizing the algorithm of Aloulou *et al* for problem $1(sa)|r_i, prec|(f_{\max} \rightarrow \max)$ [1]. Here $prec_k$ denotes precedence constraints appearing only between operations scheduled on the same machine, besides the classical flow-shop precedence constraints.

Another class of related work in the context of flexibility generation for online scheduling is linked to the concept of groups of permutable operations [6, 11], also called ordered (group) assignment [3, 27]. A group of permutable operations is a restriction of the sequential flexibility considered here in such a way that each operation is assigned to a group and there is a complete order between the groups of operations performed on the same machine. There are no precedence constraints between the operations of the same group. A pioneering work for the definition of the groups of permutable operations concept and the generation of flexible solutions has been achieved by Erschler and Roubellat [11] in the context of a job-shop problem with due dates. However no computational experiments were given to validate the practical interest of the approach. This has been achieved later but independantly by Wu *et al* [27] who define the identical ordered assignement representation and propose an approach that computes an ordered assignement in a job-shop (i.e. ordered groups of permutable operations on each machine), focusing first on resolving a critical subset of scheduling decisions. As in the work of Erschler and Roubellat [11], the principle is to allow the remaining scheduling decisions to be made dynamically in the presence of disturbances. They show through numerical experiments on the weighted tardiness job-shop that this approach is superior to the one that generated a complete solution, in the presence of small to medium disturbances.

Other heuristics have been designed to generate groups of permutable operations for general disjunctive problems [6] and multiobjective methods have been designed to find a compromise between flexibility and performance in the two-machine flowshop [12]. Artigues *et al* [3] establish the correspondence between the groups of Erschler and Roubellat [11] and the ordered assignment of Wu *et al* [27] and make a synthesis by calling this representation the ordered group assignment. They also propose a polynomial algorithm to perform the exact worst-case evaluation of an ordered group assignment. This method is based on longest path computations in a so-called worst-case graph, derived from the considered ordered group assignment. This method solves problem SP for general disjunctive problems (e.g. job-shop) where the disjunctions appear only between operations of the same group (inside each group the graph of disjunctions $e_{ij}$ is a clique) and when the precedence constraints are defined between operations of different groups.

As an illustration, the selection $\pi$ proposed for the flow-shop example yielding the 6 feasible schedules of Figure 2 with a worst-case makespan of 20 cannot be represented by groups of permutable operations. Recently Briand *et al* [7] have proposed to charaterize a set of optimal schedules for the two-machine permutation flowshop $F2|prmu|C_{\max}$ by means of interval structures. The interval structure provides a partial order which does not involve the restrictions of the concept of groups of permutable operations. Aloulou and Portmann [2] consider the single-machine scheduling problem with dynamic job arrival and total weighted tardiness and makespan as objective functions and propose a genetic algorithm to compute a flexible solution based on a partial order of the jobs.

In this paper we provide a general framework for these previous works by defining formally the problem of computing the worst-case completion times of the operations in the set of semi-active schedules compatible with a given partial order of the operations on the machines. We show that this problem is polynomially solvable for the nonpermutation flow-shop and we provide a dynamic programming algorithm to solve it.

# 4 A longest path formulation of the maximisation problem

Let $\hat{C}_i$ denote the worst case completion time of operation $i$, i.e. $\hat{C}_i = \max_{\pi\in\Pi} C_i(\pi)$. Computing $\hat{C}_i$, for each $i \in N$ solves problem SP since the objective function is a minmax function of non decreasing functions of the completion times. Recall that $\mathcal{D}$ is the set of all disjunctive arcs. Let us now consider the following Constrained Longest Path problem associated to operation $i$ ($CLP(i)$).

**Definition 1** *Given a disjunctive graph $\mathcal{G} = (V, C, D)$ and an operation $i$, problem $CLP(i)$ consists in computing the longest elementary path $L^*(0, i)$ from $0$ to $i$ in $G(\mathcal{D}) = (V, C \cup \mathcal{D})$ such that $G(L^*) = (V, C \cup L^*(0, i))$ is acyclic.*

We have the following result.

**Theorem 1** *The worst case completion time $\hat{C}_i$ is equal to the length of path $L^*(0, i)$ solution of problem $CLP(i)$.*

**Proof.** We first show that (a) $\hat{C}_i$ is the length of an elementary path $l$ from $0$ to $i$ in $G(\mathcal{D})$ and that $G(l)$ is acyclic. Let $\pi \in \Pi$ such that we have $\hat{C}_i = C_i(\pi)$. $\pi$ is the complete selection such that $\hat{C}_i$ is the length of a longest path $l$ from $0$ to $i$ in $G(\pi)$. Since $G(\pi)$ is acyclic, $l$ is elementary and since $l \subseteq \pi \cup C$, $G(l)$ is also acyclic. Since $\pi \subset \mathcal{D}$, $l$ is also an elementary path in $G(\mathcal{D}) = (V, C \cup \mathcal{D})$.

Let us show that (b) any elementary path $L$ from $0$ to $i$ in $G(\mathcal{D})$ verifying $G(L)$ is acyclic is such that there exists a feasible complete selection $\pi \in \Pi$ verifying $C \cup L \subseteq C \cup \pi$. Suppose that $L$ includes only conjunctive arcs. Then $L \subseteq C$ and (b) is verified. Suppose now that $L$ includes also disjunctive arcs. Since $L$ is elementary, we have $|L \cap e_{ij}| \leq 1$ for each disjunction $e_{ij}$. Hence $L \setminus L \cap C$ is a partial selection. Furthermore, since $G(L) = G(V, C \cup L)$ is acyclic $C \cup L$ defines a new acyclic precedence constraints graph and the disjunctive problem defined by $(V, C \cup L, D(L))$ is feasible. Hence $L \setminus L \cap C$ is included in a feasible complete selection.

From (b) it follows that the length of any elementary path $L$ from $0$ to $i$, verifying $G(L)$ is acyclic, is less or equal than $\hat{C}_i$. We proved in (a) that there exists an elementary path $l$, such that $G(l)$ is acyclic, with a length equal to $\hat{C}_i$. Hence, $\hat{C}_i$ is the length of $CLP(i)$-solution. ■

Note that in the general case, problem $CLP(i)$ may be not easy to solve since it admits as a particular case the search for the longest elementary path in a graph with positive length cycles. This problem is known to be NP-hard for general graphs [13]. In Figure 3 we illustrate the problem and the necessity of the no-cycling condition in definition 1 for a job-shop with 2 machines, 3 jobs and no release dates. Operations 1,4 and 5 are assigned to the first machine and operations 2,3 and 6 are assigned to the second machine. Structural precedence constraints are $(1, 2)$, $(3, 4)$ and $(5, 6)$. $(3, 2)$ and $(1, 4)$ are additional precedence constraints. The longest elementary path from 0 to 7 is displayed in bold. Such a path is infeasible since it induces a cycle with precedence constraints $(3, 4)$. Hence, it is not a solution of $CLP(7)$.

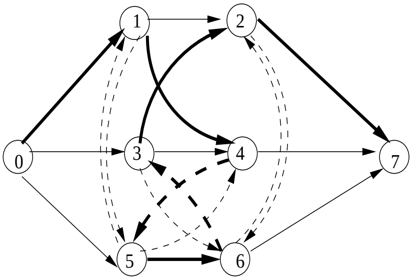



Figure 3: A job-shop example and an infeasible elementary longest path

The following proposition shows on the opposite that the problem is simplified in a nonpermutation flow-shop context, since the no cycling condition is not necessary.

**Proposition 1** *Given a disjunctive graph $\mathcal{G} = (V, C, D)$ of a nonpermutation flow-shop with additional precedence constraints and an operation $i$, if $L(0, i)$ is an elementary path from $0$ to $i$ in $G(\mathcal{D}) = (V, C \cup \mathcal{D})$ then $G(L) = (V, C \cup L)$ is acyclic.*

**Proof.** Due to the flow-shop structure, the strongly connected components of $G(\mathcal{D})$ include only operations assigned to the same machine. Hence any elementary cycle of $G(\mathcal{D})$ involves only operations assigned to the same machine. Let $L$ denote an elementary path in $G(\mathcal{D})$. By definition $L$ is acyclic and no cycle can be created by adding structural precedence constraints to $L$. ∎

We show in next Section that the problem is polynomially solvable in the flow-shop context.

# 5 A polynomial algorithm for the nonpermutation flow-shop case

In this section, we consider the nonpermutation flow-shop problem with operation release dates and additional precedence constraints appearing only between operations scheduled on the same machine, as in the example presented is section 2. In this case, any sequence of operations compatible with the precedence constraints of machines yields a feasible complete selection. For each operation $j$, let $j^-$ denote its job predecessor. We assume that if $j$ is the first operation of its job, then $j^-$ is a dummy operation denoted $j^0$. Let $\Gamma_j^-$ (resp. $\Gamma_j^+$) denote the set of operations that must be scheduled before (resp. after) $j$ on machine $m_j$. Let $I_j$ denote the set of operations of machine $m_j$ that are not linked to $j$ with any precedence constraint. We have

$$\Gamma_j^- = \{i \neq j | m_i = m_j \text{ and there is a path from } i \text{ to } j \text{ in } G = (V, C)\} \tag{1}$$

$$\Gamma_j^+ = \{i \neq j | m_i = m_j \text{ and there is a path from } j \text{ to } i \text{ in } G = (V, C)\} \tag{2}$$

$$I_j = \{i \neq j | m_i = m_j, i \notin \Gamma_j^- \text{ and } j \notin \Gamma_i^-\} \tag{3}$$

Let us define $\hat{C}_{j^0} = r_j$. We have the following result.

**Lemma 1** *The worst case completion time of any operation $j$ is given by*

$$\hat{C}_j = p_j + \max \begin{cases} r_j, & (a) \\ \hat{C}_{j^-}, & (b) \\ \max\limits_{i \in I_j \cup \Gamma_j^-} \{\max(r_i, \hat{C}_{i^-}) + \sum\limits_{x \in I_j \cup \Gamma_j^- \setminus \Gamma_i^-} p_x\} & (c) \end{cases}$$

**Proof.**

Consider machine 1 and an operation $j$ to be executed on this machine. We have $\hat{C}_{j^-} = \hat{C}_{j^0} = r_j$, hence terms (a) and (b) are redundant. Denote by $S$ the semi-active schedule in which $C_j(S) = \hat{C}_j$ and the block $B$ of operations consecutive on machine 1, ending with $j$, is such that there is no idle time between any two consecutive operations in $B$ and $B$ is of maximal size. $B$ always exists since we have at least $j \in B$. If $B = \{j\}$, then the starting time of $j$, $S_j$, is such that $S_j = r_j$ and (a) is verified.

If $|B| > 1$, then we have $S_j \geq r_j$. Let $i$ be the first operation of block $B$. $i$ is not a machine successor of $j$ and $i \in I_j \cup \Gamma_j^-$. Similarly, by definition all operations inside $B$, except $j$ itself, belong to $I_j \cup \Gamma_j^- \setminus \Gamma_i^-$ (they cannot be machine predecessors of $i$). Let us now consider the operations scheduled before $i$ on machine 1. Let $x$ denote the operation scheduled at the largest position before $i$ such that $x \notin \Gamma_i^-$. This operation could be inserted right after $i$. The obtained schedule is semi-active and the the completion time of $j$ increases, which contradicts the maximality of $C_j(S)$. Hence all operations scheduled before $i$ are in $\Gamma_i^-$. This implies that all operations of $I_i \cup \Gamma_j^- \setminus \Gamma_i^-$ are scheduled after $i$. Conversely, suppose that $x$ is the operation scheduled at the smallest position after $j$ such that $x$ is not a successor of $j$, i.e. $x \notin \Gamma_j^+$. This operation could be inserted right before $j$ providing a new semi-active schedule in which the start time of $j$ increases, which contradicts the maximality of $C_j(S)$. Hence all operations scheduled after $j$ are successors of $j$. It follows that if $S_j > r_i$ then

$$\hat{C}_j \leq \max_{i \in I_j \cup \Gamma_j^-} \{r_i + \sum_{x \in I_j \cup \Gamma_j^- \setminus \Gamma_i^-} p_x\} + p_j. \tag{4}$$

Can we have $i \in I_j \cup \Gamma_j^-$ such that $\hat{C}_j < r_i + \sum_{x \in I_j \cup \Gamma_j^- \setminus \Gamma_i^-} p_x + p_j$?

Suppose that $i$ is such an operation. It is possible to build a feasible semi-active schedule in which all the operations before $i$ are machine predecessors of $i$ and the operations after $j$ are only machine successors of $j$. This can be made by scheduling the operations of $\Gamma_i^-$ in an order compatible with the precedence constraints within this set, then $i$, then the operations of $I_i \cup \Gamma_j^+$ in an order compatible with the precedence constraints within this set, then operation $j$, then the operations of $\Gamma_j^+$ in an order compatible with the precedence constraints within this set. The operations on machine 2 can be scheduled in any order compatible with the precedence constraints of machine 2, and so on. The obtained schedule $S$ is semi-active and we have $C_j \geq r_i + \sum_{x \in I_j \cup \Gamma_j^- \setminus \Gamma_i^-} p_x + p_j$. Hence (4) is verified to equality.

We can use the similar arguments to prove the result for any machine $k > 1$.

Consider a machine $k$ and an operation to be executed on this machine. Let $S$ be a semi-active schedule in which $C_j(S) = \hat{C}_j$ and the block $B$ of operations consecutive on machine $k$, ending with $j$, is such that there is no idle time between any two consecutive operations in $B$ and $B$ is of maximal size. If $|B| = 1$ then we have either $C_j(S) = r_j + p_j$ or $C_j(S)$ is set by $C_{j^-}$. To maximize this value we have $C_j(S) = \hat{C}_{j^-} + p_j$.

If $|B| > 1$ we can also state that an operation $i \in I_j \cup \Gamma_j^-$ starts the block with $S_i = r_i$ or $S_i = C_{i^-}$. With similar arguments as for machine 1, we prove that all operations of the set $I_j \cup \Gamma_j^- \setminus \Gamma_i^-$ are in the block. Furthermore if $S_i > r_i$ then we have $S_j = C_{i^-} = \hat{C}_{i^-}$ to have $\hat{C}_j$ maximal. Last we can show that for any operation $i \in I_j \cup \Gamma_j^-$, we can build a feasible semi-active schedule in which all the operations before $i$ are machine predecessors of $i$ and the operations after $j$ are machine successors of $j$ and $S_i = \max(r_i, \hat{C}_{i^-})$. This achieves the proof. ■

Let $\nu = n/m$ be the number of jobs. Due to lemma 1, we have the following result.

**Theorem 2** *Problem $F(sa)|r_i, prec_k|(f_{\max} \rightarrow \max)$ can be solved in $O(m\nu^3)$ times if each function $f_i$ is computable in $O(1)$ time.*

**Proof.** Once sets $\Gamma_j^-$ and $I_j$ are built for each operation $j$, all worst-case completion times can be computed trivially via the proposed recursion by dynamic programming in $O(m\nu^3)$. ■

In the illustrative example of section 2, the worst case completion times are given (in the order of their computation) by $\hat{C}_1 = r_1 + p_1 = 1$ (a), $\hat{C}_3 = r_5 + p_5 + p_3 = 8$ (c), $\hat{C}_5 = r_1 + p_1 + p_3 + p_5 = 7$ (c), $\hat{C}_7 = r_5 + p_5 + p_3 + p_7 = 14$ (c), $\hat{C}_2 = \hat{C}_1 + p_2 = 7$ (b), $\hat{C}_4 = \hat{C}_7 + p_8 + p_4 = 20$ (c), $\hat{C}_6 = \hat{C}_3 + p_4 + p_6 = 19$ (c), $\hat{C}_8 = \hat{C}_3 + p_4 + p_6 + p_8 = 20$ (c).

# 6 Flexible solutions for the flow-shop problem with a common due-date

In this Section, we show how flexible solutions can be computed for a flow-shop problem with a common due date. Contrarily to most references encountered in the litterature we do not restrict the set of schedules to permutation schedules, where the order of the jobs has to be identical on all machines [17, 24, 26]. Formally, the problem solved in this section can be denoted $F|\tilde{d}_i = d|-$, i.e. a flow-shop in which all jobs have a common due-date and the objective is to obtain a feasible schedule. Solving this problem can also solve the makespan minimization problem $F||C_{\max}$.

In this Section we solve problem $F|\tilde{d}_i = d|-$ through branch and bound. Inside the branch and bound we incorporate worst case completion time computations to issue a flexible solution rather than a single schedule.

In the remaining Section, we briefly give the elements of the branch and bound, all borrowed from previous studies: the branching scheme (Section 6.1), the constraint propagation algorithms used at each node to sharpen the operation time windows (Section 6.2), the heuristic used at each node to try to find a feasible solution (Section 6.3). In Section 6.4 we explain how the worst-case computation methods have been integrated in the branch and bound scheme. Last, Section 6.5 provides computational experiments on standard flow-shop instances.

## 6.1 Branching Scheme

The proposed branching scheme is based on the disjunctive graph. At each node the disjunctive graph is updated through the last branching decisions. The branching rules are based on the relative ordering of the operations assigned to the same machines. At each node a machine is selected and a child node is generated for each operation candidate for being scheduled next on the machine. The machine is selected as the one on which the operation with the smallest release date $i^*$ is assigned. The candidates for being scheduled first on this machine are the operations with a release date not greater than the earliest completion time of $i^*$. All disjunctive arcs issued from the selected operation are then orientated as outgoing arcs for this operation. Hence at each node, a disjunctive graph $\mathcal{G} = (V, C, D)$ is defined where $C$ includes the structural and the additional resource precedence constraints. The tree is explored by depth-first search.

## 6.2 Constraint propagation

The common due date $d$ allows to compute a time window $[r_i, d_i]$ for each operation $i \in N$. Constraint propagation algorithms are used at each node to maintain the time window as tight as possible, to detect implied precedence constraints and to prune the node if inconsistency is proven. The release times $r_i$ and the due dates $d_i$ are first computed with forward and backward longest path computations in $(V, C)$. Time window tightening, precedence constraint detection and consistency checking are performed by the disjunctive constraint propagation and edge-finding techniques. For a precise description of these techniques, we refer to [8, 5].

Furthermore at the root node initial time windows are computed by the shaving technique [19, 9] which consists in running the above-referred constraint propagation algorithms after setting the start time of an operation to its release date (or to its due date). If inconsistency is proven, the tentative value can be removed from the time window. Such a technique has been proven very useful for flow-shop problems [20].

## 6.3 Heuristic

In the case where the current node has not been pruned by constraint propagation, a heuristic is used to find a feasible solution. The heuristic is simply based on the application of the well-known priority-rule based active and non-delay constructive algorithms [4]. At each node, we apply 4 times the non-delay scheduling algorithms and 4 times the active scheduling algorithm. The 4 used priority rules are the minimal earliest possible start ($r_i$), the minimal lastest start ($d_i - p_i$) and the randomized version of these rules where another operation then the one determined by the rule is selected with a low probability.

When no feasible solution has been found, the solution with the lowest makespan is kept. Each time one of the 8 constructive methods improves the best known solution in terms of makespan, an intensification phase is applied by running $H1$ times the randomized version of the priority rule that yielded the improvement with both active and non-delay algorithms. This amounts to a basic neighborhood search of a further improving solution

## 6.4 Integration of worst-case completion times

The branch and bound additionnaly uses worst-case completion time computations. Let $\mathcal{G} = (V, C, D)$ denote a disjunctive graph which is worst-case feasible, i.e. for which the worst case completion time $\hat{C}_i$ of each operation verifies $\hat{C}_i \leq d_i$. Since $\mathcal{G}$ is feasible in the worst-case, we may prune the node and keep $\mathcal{G}$ as a flexible solution. Although at this time, problem $F|\tilde{d}_i|-$ is solved, the search process continues however to find further flexible solutions. Note that we aim at finding a flexible solution with a maximal number of disjunctions $|D|$ which represent the greatest amount of decisions let to the decision maker.

Each time a feasible solution is found (or the node is worst-case feasible), we run a heuristic (WCH) returning a flexible solution from a given worst-case feasible disjunctive graph $\mathcal{G}$ described as follows. The disjunctive-graph is modified such that it becomes minimally worst-case feasible, i.e. it does not include any orientated disjunctive arc $(i, j)$ such that $(V, C \setminus (i, j), D \cup e_{ij})$ is worst case feasible. The heuristic traverses all arcs $(i, j)$ of $C$ and check whether the latter property is verified for $(i, j)$. If an arc $(i, j)$ such that $(V, C \setminus (i, j), D \cup e_{ij})$ is worst case feasible is found, $(i, j)$ is removed from $C$ and $e_{ij}$ is added to $D$. The process is iterated until $\mathcal{G}$ becomes minimally worst-case feasible. Note that this process increases the size of the solution space represented by $\mathcal{G}$. Only those arcs $(i, j)$ such that there is no other path from $i$ to $j$ are considered for being removed. Otherwise, removing $(i, j)$ does not remove any precedence constraint.

The order in which arcs are selected for being removed from $C$ determines the resulting disjunctive graph. Hence several orders may result in different minimally worst-case feasible graphs. We call the above procedure with 100 randomly generated arc orders for each encountered feasible or worst-case feasible node. The branch and bounds stops when this process has been applied in turn 100 times, which correspond to a total of 10000 calls to the WCH procedure, or there is no more node to develop.

Recall that the branch and bound is an exact method w.r.t problem $F|\tilde{d}_i = d|-$. The number of disjunctions of the flexible solution is heuristically maximized only.

### 6.5 Experimental comparison

In this Section we give the performance of the branch and bound on flowshop instances issued from the litterature.

Because of the difficulty of the non permutation flow-shop problem problem, we have modified the smallest instances designed by Taillard [23], which originally comprise 10 problems with 20 jobs and 5 machines, to keep only the 10 first jobs in each. All programs have been coded in C++ and run on an AMD64 archicture under Linux. Cplex 9.0 was used to solve the LP relaxations and the MILP problems. Parameters H1 was set to 50000 iterations.

We have made 2 series of experiments, one with the common due date of each instance set to the optimal makespan and the other one with the common due date set to the optimal makespan augmented by 5%. The results are given in Table 1 for the instances with the tight common due date and in Table 2 for the instances with the loose common due date. In both tables, we give for each instance the number of jobs, the number of machines, the minimal makespan, the common due date, the largest obtained number of disjunctions of the flexible solution (also expressed in percentage of the total number of disjunctions, equal to 225). We also provide the numbers of nodes and the CPU times in seconds needed to obtain the first feasible solution and the numbers of nodes and CPU times needed to obtain the flexible solution. The results show that in both cases, flexible solutions are exhibited with a reasonable amount of additional CPU time. As expected the flexibility is higher for the instances with a loose common due-date but significant flexibility is also generally obtained for the instances with the tight due date.

## 7 Conclusion

In this paper, we proposed a longest path formulation of the problem of evaluating the worst case performance of flexible solutions in disjunctive scheduling with minmax regular objective function. A flexible solution is defined by an operation partial order on each machine. We proved that this problem is polynomial in the special case of the

Table 1: Results on instances with a tight common due date

| P | #jobs | $m$ | $C^*_{\max}$ | $d$ | $\|D\|$ | #nodes 1st | CPU 1st | #nodes | CPU |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 10 | 5 | 767 | 767 | 3 (1.3%) | 5286 | 18 | 6614 | 21 |
| 2 | 10 | 5 | 763 | 763 | 6 (2.6%) | 7 | 0 | 1660 | 40 |
| 3 | 10 | 5 | 691 | 691 | 7 (3.1%) | 12 | 0 | 837 | 52 |
| 4 | 10 | 5 | 813 | 813 | 3 (1.3%) | 1 | 0 | 33 | 11 |
| 5 | 10 | 5 | 731 | 731 | 8 (3.5%) | 4666 | 12 | 7295 | 118 |
| 6 | 10 | 5 | 749 | 749 | 9 (4%) | 267 | 0 | 6373 | 20 |
| 7 | 10 | 5 | 741 | 741 | 11(4.8%) | 1676 | 10 | 3254 | 16 |
| 8 | 10 | 5 | 717 | 717 | 4 (1.7%) | 58 | 6 | 1026 | 12 |
| 9 | 10 | 5 | 687 | 687 | 7 (3.1%) | 1 | 0 | 62 | 27 |
| 10 | 10 | 5 | 762 | 762 | 17(7.5%) | 14 | 0 | 499 | 150 |

Table 2: Results on instances with a loose common due date

| P | #jobs | $m$ | $C^*_{\max}$ | $d$ | $\|D\|$ | #nodes 1st | CPU 1st | #nodes | CPU |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 10 | 5 | 767 | 805 | 14 (6.2%) | 180 | 0 | 3500 | 146 |
| 2 | 10 | 5 | 763 | 801 | 9 (4%) | 1 | 0 | 758 | 70 |
| 3 | 10 | 5 | 691 | 725 | 11 (4.8%) | 1 | 0 | 2778 | 105 |
| 4 | 10 | 5 | 813 | 853 | 10 (4.4%) | 1 | 0 | 4805 | 109 |
| 5 | 10 | 5 | 731 | 767 | 11 (4.8%) | 2 | 0 | 34104 | 177 |
| 6 | 10 | 5 | 749 | 786 | 16 (7.1%) | 1 | 0 | 3708 | 162 |
| 7 | 10 | 5 | 741 | 778 | 22 (9.7%) | 21 | 0 | 13748 | 220 |
| 8 | 10 | 5 | 717 | 752 | 9 (4%) | 16 | 0 | 704 | 90 |
| 9 | 10 | 5 | 687 | 721 | 21 (9.3%) | 4 | 0 | 7807 | 186 |
| 10 | 10 | 5 | 762 | 800 | 19 (8.44%) | 1 | 0 | 515 | 193 |

flow-shop problem with release dates and additional precedence constraints between operations scheduled on the same machine. We used the worst case computation method to generate flexible solutions for a flow-shop problem with a common due date. We show that flexible solution with a significant flexibility are obtained with a reasonable computational overhead with partial solutions leaving unselected up to 10% of the disjunctions when the due date is loose and up to 7% of the disjunction when the due date is tight.

Besides their interest for on-line decision support, flexible solutions could also be used in bricriteria scheduling as a support to $\epsilon$-constraint method [10, 16, 25]. Indeed, once a set of schedules achieving a required worst-case value on the first (regular minmax) criterion, the optimal solution for the second criterion can be searched on this set without considering any constraint on the first criterion, as underlined by Gupta and Stafford in [14], about the work of [7].

Another work of interest would be to focus on extensions of the worst-case performance evaluation procedure to more general problems. Unfortunately, extending this approach to the job shop is not trivial. A way to solve it is to study the complexity of the problem of finding the constrained longest elementary path in the disjunctive graph of the job-shop problem. It is also of great interest to investigate maximization problems with total (weighted) flow time as objective function. Indeed, the flow-shop problem $F(sa)|r_j|(\sum C_j \to \max)$ is open whereas the single machine $1(sa)|r_j|(\sum w_j C_j \to \max)$ is polynomially solvable [1].

# Acknowledgments

The authors are very grateful to Eric Sanlaville for the proposed suggestions to improve the presentation and the content of the paper.

# References

[1] M. Aloulou, M. Kovalyov, and M.C. Portmann. Maximization in single machine scheduling. *Annals of Operations Research*, 129:21–32, 2004.

[2] M.A. Aloulou and M.-C. Portmann. An efficient proactive-reactive scheduling approach to hedge against shop floor disturbance. In G. Kendall, E.K. Burke, S. Petrovic, and M. Gendreau, editors, *Multidisciplinary Scheduling: Theory and Applications 1st International Conference, MISTA '03 Nottingham, UK, 13-15 August 2003. Selected Papers*, pages 223–246. Elsevier, 2005.

[3] C. Artigues, J.C. Billaut, and C. Esswein. Maximization of solution flexibility for robust shop scheduling. *European Journal of Operational Research*, 165(2):314–328, 2005.

[4] K. R. Baker. *Introduction to sequencing and scheduling*. Wiley, 1974.

[5] Ph. Baptiste, C. Le Pape, and W. Nuijten. *Constraint-Based Scheduling*. Kluwer Academic Publishers, 2001.

[6] J.C. Billaut and F. Roubellat. A new method for workshop real time scheduling. *International Journal of Production Research*, 34(6):1555–1579, 1996.

[7] C. Briand, H.T. La, and J. Erschler. A new sufficient condition of optimality for the two-machine flowshop problem. *European Journal of Operational Research*, 169(3):712–722, 2006.

[8] J. Carlier and E. Pinson. An algorithm for solving the job-shop problem. *Management Science*, 35(2), 1989.

[9] J. Carlier and E. Pinson. Adjustment of heads and tails for the job-shop problem. *European Journal of Operational Research*, 78:146161, 1994.

[10] V. Chankong and Y. Haimes. *Multiobjective decision making theory and methodology*. Elsevier, 1983.

[11] J. Erschler and F. Roubellat. An approach for real time scheduling for activities with time and resource constraints. In R. Slowinski and J. Weglarz, editors, *Advances in project scheduling*. Elsevier, 1989.

[12] C. Esswein, J.C. Billaut, and V. Strusevich. Two-machine shop scheduling: Compromise between flexibility and makespan value. *European Journal of Operational Research*, 167(3):796–809, 2005.

[13] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[14] J.N.D. Gupta and E. F. Stafford. Flowshop scheduling research after five decades. *European Journal of Operational Research*, 169(3):699–711, 2006.

[15] W. Herroelen and R. Leus. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165(2):289–306, 2005.

[16] H. Hoogeveen. Multicriteria scheduling. *European Journal of Operational Research*, 167(3):592–623, 2005.

[17] J. Lemesre, C. Dhaenens, and E.G. Talbi. An exact parallel method for a bi-objective permutation flowshop problem. *European Journal of Operational Research*, 177(3):1641–1655, 2007.

[18] J. K. Lenstra, A. H. G. Rinnooy Kan, and P. Brucker. Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, 1, 1977.

[19] P. Martin and D.B. Shmoys. A new approach to computing optimal schedules for the job-shop scheduling problem. In *5th International IPCO Conference*, pages 389–403, 1996.

[20] L. Peridy, E. Pinson, and D. Rivreau. Enhanced disjunctive elimination rules for the flow-shop and permutation flow-shop problems. In *6th International Workshop on Project Management and Scheduling*, Istanbul, 1998.

[21] M. E. Posner. Reducibility among wighted completion time scheduling problems. *Annals of Operations Research*, pages 91–101, 1990.

[22] B. Roy and B. Sussmann. Les problèmes d'ordonnancement avec contraintes disjonctives, 1964. D.S. vol. 9, SEMA, Paris, France.

[23] E.D. Taillard. Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64:278–285, 1993. available at http://ina2.eivd.ch/collaborateurs/etd/problemes.dir/ordonnancement.dir/ordonnancement.html (visited on April, 3 2006).

[24] M. F. Tasgetiren, Y.-C. Liang, M. Sevkli, and G. Gencyilmaz. A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. *European Journal of Operational Research*, in press.

[25] V. T'Kindt and J.-C. Billaut. *Multicriteria Scheduling*. Springer, 2002.

[26] V. T'Kindt, J.N.D. Gupta, and J.-C. Billaut. Two-machine flowshop scheduling with a secondary criterion. *Computers and Operations Research*, 30:505–526, 2003.

[27] S.D. Wu, E.S. Byeon, and R.H. Storer. A graph-theoretic decomposition of the job-shop scheduling problem to achieve scheduling robustness. *Operations Research*, 47(1):113–124, 1999.

# A note on the role of robustness analysis in decision-aiding processes

Luis C. Dias [*]

**Résumé**

L'analyse de robustesse est, de façon croissante, une réponse populaire face aux difficultés à établir la valeur des paramètres des modèles d'aide à la décision, soit pour des décisions individuelles, soit pour des décisions de groupe. Cette note présente une discussion de trois rôles possibles de l'analyse de robustesse dans un processus décisionnel. En considérant les difficultés pour fixer la valeur des paramètres concernant les préférences dans les procédures, nous examinons les avantages potentiels du recours à l'analyse de robustesse pendant toute la durée du processus décisionnel en tant qu'outil pour guider le processus ; ceci est l'idée sous-jacente aux logiciels tels que VIP Analysis ou IRIS et à leurs extensions aux décisions multi-acteur. Notre discussion est centrée sur la recherche de conclusions robustes plutôt que sur celle de solutions robustes, soulignant ainsi le rôle de la modélisation comme processus constructif d'apprentissage.

**Mots-clefs :** Conclusion robuste ; paramètres concernant préférences ; décision de groupe ; processus décisionnel**.**

**Abstract**

Robustness analysis is an increasingly popular response to the difficulties in setting the parameter values of decision-aiding models, for individual or group decisions. This note discusses three possible roles for robustness analysis in a decision-aiding process. Considering the difficulties of eliciting values for preference-related parameters, we will examine the potential benefits of using robustness analysis throughout the whole decision process as a tool to guide that process, which is the idea behind tools such as VIP Analysis or IRIS and their

[*] INESC Coimbra / Faculty of Economics, University of Coimbra, Av Dias da Silva 165, 3004-512 Coimbra, Portugal. { LDias@inescc.pt }

extensions for group decision-making. Emphasis is placed on the concept of robust conclusion, rather than robust solution, as well as on the idea of modeling as a constructive learning process.

**Key Words:** robust conclusion; preference-related parameters; group decision; decision process.

# 1 Introduction

Formal models used in decision-aiding processes are subject to many sources of uncertainty, imprecision, and ignorance (Roy 1989). We will consider that a formal model is defined by a structure and a combination of parameter values. For instance, the structure of a linear program consists of a linear function to maximize (or minimize) and a set of linear constraints; its parameters are the cost coefficients, the technical coefficients, and the right-hand-side values. A multicriteria decision-aiding model can be structured as an ELECTRE model, a MAUT model, or another type of structure; its parameters are related to the performance of each alternative on each criterion, the criteria weights, etc. For a given model structure, different combinations for the parameter values will lead to different model "versions" (to use the expression recently proposed by Roy (2003, 2007)).

It is well known that setting technical and economical parameter values is often problematical: instruments and statistics can be imprecise (e.g., tolerances for precision in measurement, confidence intervals in statistics), measurement can be arbitrary and subjective (e.g., measuring noise pollution or a firm's performance), some information may be controversial or contradictory (e.g. data from clinical trials), let alone uncertainties about the future.

When considering multi-criteria decision aiding, as we wish to do, the models also incorporate parameters related to the preferences of the Decision Maker (DM). Eliciting parameter values about preferences is also problematical. In cognitive terms, the parameters are artifacts whose semantic may be difficult to understand for the DM, not to mention biases related to the way questions are posed. For him or her, value judgments are naturally easier to express through words than through numbers. Furthermore, preferences may evolve, as they are often unstable outcomes of unresolved internal conflicts in the DM's mind. Adding to these fundamental difficulties, other constraints of a more pragmatic nature may be present, e.g., the DM is reluctant to divulge precise parameter values about his/her preferences in public, or his/her time and patience is rather limited.

Moreover, we often need to address the concerns of a group of actors, rather than a single DM. The above mentioned difficulties of fixing preference-related parameter values are still present, if not reinforced by the diversity of judgments and subjective perceptions of what is at stake. In such cases, the existence of "hidden agendas" may hinder an open discussion about parameter values. Even in case of consensus, one must be aware of phenomena such as groupthink (Janis 1972).

Robustness Analysis (RA) is motivated by these difficulties in setting the parameter values of decision-aiding models, i.e., the difficulties of isolating a single *model version*[1] to serve as a basis to recommend a decision. The general idea behind RA approaches is to accept multiple model versions and to try to progress in the decision-aiding process despite the diversity of versions, possibly to identify a solution that is seen as being good (or at least as being acceptable) in every model version (or at least in most of the versions). As a matter of fact, we may find a reference to "robustness" in this sense in one of the pioneering books on Operational Research (OR), which described a famous OR problem referring to the need of solutions «which are robust in a quite large variety of circumstances» (Beer, 1966: p.44). Although we will not address this topic, it should be noted that the *robustness concern* (Roy 2007) may also be related to different versions of the method or algorithm that is applied to derive solutions from the model version or versions (see also Vincke, 1999).

In this note we intend to discuss the role of RA in decision-aiding processes in general, both for single-actor and multi-actor decisions. We will emphasize the word *decision* as an activity or process (as in the sentence "it was a difficult decision"), rather than *decision* as an object or solution (as in the sentence "it was a good decision"). Naturally, given the multiple meanings attached to the name *RA*, we will start by briefly reviewing some of these multiple concepts of RA, and we will propose a broad tentative definition. Then, we discuss three possible roles for RA in a decision- aiding process, emphasising the role underlying the use of RA in tools such as VIP Analysis (Dias and Clímaco 2000) and IRIS (Dias and Mousseau 2003), as well as its potential role in aiding group decisions.

# 2 Concepts of Robustness Analysis

Sensitivity Analysis (SA) is a traditional answer to the difficulties in setting the model's parameter values. In model building it indicates which parameters contribute more to the variance of the outputs. In optimization, it indicates how much the parameters may vary without changing some conclusion of interest. RA is often seen as a reverse perspective of SA, but that would depend on the notion of RA that is being considered. Indeed, we may find multiple perspectives about the concept (Roy, 2002).

[1] In this issue, Roy (2007) suggests the designation "problem formulation version". The designation *model version* or *problem formulation version* seems more adequate than the designation *problem version*, which Roy (2003) had proposed earlier, to emphasize that it is the model that will vary, not the problem.

To Rosenhead (2001), RA is used to choose one action (a first choice in a succession of choices) that is flexible enough as to leave many good options regarding the choices to be made in the future. Kouvelis and Yu (1997) define robust solution to an optimization problem as the one which has the best performance in its worst case (e.g., max-min rule), which could be replaced by the criterion proposed by Aloulou et al. (2005), for instance. Mulvey et al. (1995) differentiate between the quality of solution-robust, for a solution whose value is always near the optimal one for every acceptable model version, from the quality of model-robust, for a solution which is always feasible or almost feasible for every version. Sevaux and Sörensen (2004) introduce a concept of solution robustness meaning the solution (a plan) does not change much in optimization programs that are to be repeated regularly. More generally, Hites et al. (2003) recommend a multicriteria evaluation of solution robustness. For more references and a classification of research work on the concept of robustness, see in this issue (Roy 2007).

The perspectives that are nearer to the reverse of SA are those appearing in (Roy 1998; Roy and Bouyssou, 1993), defining robust conclusion as an assertion that is valid for the set of results compatible with the different model versions envisaged, and Vincke's (1999) definition of robust solution as one that is always near (or that does not contradict) any other solution obtainable using an acceptable version (Vincke has also introduced the notion of robust method). Roy (1998) introduced the notions of approximately robust conclusion (if it is true "almost everywhere") and pseudo-robust conclusion (if not perfectly formalized). It may also be of interest to distinguish absolute vs. relative conclusions, as well as to distinguish between conclusions about the intermediate results vs. the final result of a method (e.g., see Dias and Clímaco 2002).

As a working definition in this paper, we consider RA in Operational Research / Decision Aiding as an activity whose purpose is to identify what conclusions are robust for a set of model versions (and, in some cases, a set of method/algorithm versions). Taken broadly, this definition encompasses the activity of defining a set of model versions, the procedures to find out what is robust (or to test if a conclusion is robust), and the interaction with the DM(s). It should be noted that this emphasis on conclusions is more general than an emphasis on solutions. Indeed, any assertion about a robust solution (whatever the definition) can be rephrased as a robust conclusion involving that solution. The following assertions are examples of what could be robust solutions, following the classification introduced by Dias and Clímaco (1999):

- An *absolute robust conclusion* is an assertion intrinsic to one of the actions or solutions, valid for every model version. For instance, "the value of action $x$ is greater than 0.7", or "the cost of path $y$ on a network does not exceed 10".

- A *(relative) binary robust conclusion* is an assertion that relates a pair of actions or solutions, valid for every model version. For instance, "solution $x$ is better than solution *y*" or "action *x* outranks action *y* with credibility greater than 0.7" are possible binary robust conclusions.

- A *(relative) unary robust conclusion* is an assertion concerning one action or solution but relative the remaining actions or solutions, valid for every model version. For instance, "action *x* is among the three top actions in a ranking", or "*x* is the solution that achieves lowest cost in its worst scenario (version)" are possible unary robust conclusions.

The adverb *partly* can be used to extend this classification to characterize conclusions that are valid for only a proportion of the model versions (this encompasses both finite and infinite, yet bounded, sets of versions). For instance, "solution *x* is better than solution *y* for 99% of the versions" could be an example of a binary partly robust conclusion, and "among all solutions, the proportion of versions where the solution value is higher than 0.8 is 95% for solution *x*" could be an example of a unary partly robust conclusion.

Usually, the model versions are considered to be equally acceptable, without attempting to attribute different degrees of probability (or possibility, or importance…) to different versions (a kind of "meta-model"). Indeed, an attempt to distinguish between different model versions in terms of their importance or plausibility would bring new difficulties to surmount: new parameters would have to be introduced (probability distributions, or other), and a model that was simple might become too complex for being used by the DM in a transparent manner. Furthermore, it should be noted that some "meta-models" are inadequate to some types of parameters, e.g., it would be unnatural to consider criteria weights as random variables. For such reasons, the most common option is to consider a set defining the model versions that will be analyzed, considering them as equally valid (this other kind of "meta-model" is thus a set of versions without additional information / parameters).

# 3 Roles for Robustness Analysis

The role of RA in decision aiding does not seem to have been much discussed so far. In an attempt to classify most of the proposed RA approaches, we could separate them according to their placement (ex-ante vs. ex-post) with respect to using a method to obtain a solution.

One of the possibilities is to consider RA as an ex-ante concern, which amounts to imbed this concern in a model to be analysed. In these cases, usually optimization problems, a model is built and an algorithm is used to obtain a solution that is robust according to some pre-specified criterion. The obtained solution will be optimal with respect to that criterion (e.g., it minimizes the maximum cost or the maximum regret). It may even happen that the optimal solution for the RA-imbedded model might never have been optimal for any of the versions considered for the original model. Examples of these approaches are, e.g., (Kouvelis and Yu, 1997; Mulvey et al. 1995). An approach that seems particularly promising is to use several criteria rather than a single one to be optimized (as suggested by Hites et al., 2003), possibly using tools analogous to those employed in interactive multiobjective programming to explore the set of efficient solutions. Another type of approach considering RA ex-ante is to identify a subset of robust solutions (possibly empty) according to some robustness criterion (e.g., Aloulou et al., 2005, or the proposals of Roy (2007) in this issue), rather than a single optimally robust solution.

A second possibility is to consider RA as an ex-post concern, substituting or complementing SA, to assess how robust is a solution derived from a decision-aiding process and to supply additional robust conclusions. Arguably, the first example of this type of approach is found in (Roy and Bouyssou, 1993). Such an approach may be useful to question the validity of the recommendation and how its evaluation might change from version to version, possibly identifying its limits and enriching the information that may be provided to the DM. For instance, rather than saying that $x$ is the best alternative in a choice problem, one may inform the DM that all alternatives are outranked by either $x$ or $y$, explaining what are the main differences between the versions that favour $x$ and those favouring $y$, and adding that $y$ is always a relatively good choice, while there are versions where $x$ receives a poor evaluation. In the case of an optimization problem, after finding a solution that is apparently the best one, the DM may wish to compute how bad it can become for some extreme model versions to be characterized as part of this analysis.

Before discussing a third possibility, we may note that for the approaches we mentioned before the set of versions is considered to have been defined a priori. As Roy (2002) notes, this may cause a dilemma between the wish to take into account every conceivable version and the wish to obtain some useful conclusions. It is perhaps because of this potential dilemma that Roy (1998) had earlier proposed the notion of approximately robust conclusion: a formal assertion that is verified for all the versions, except a few ones, considered negligible.

When we consider preference-related parameters, a third possibility is based on the idea of using RA, i.e., identifying what conclusions are robust for a set of model versions, throughout the whole decision process as a tool to guide that process. At the outset of the process, the participants will be open to accept a wide set of model versions, which they will try to progressively reduce as the process advances. Instead of a process that starts with an elicitation of parameter values, followed by the computation of a solution and an ex-post RA, we will have a reiteration of elicitation and RA phases.

In elicitation phases, the DM will be questioned about parameter values, without requesting precise numbers (e.g., the answer can be an interval, or a comparison relation between two parameters). The elicitation may even proceed indirectly, as in aggregation-disaggregation approaches (Jacquet-Lagrèze and Siskos 2001), where the DM may indicate results that the model should reproduce. The DM's answers to the elicitation phases will be used to constrain the set of versions considered in RA phases. In the first elicitation phases the more difficult elicitation questions may be avoided, allowing the DM to learn about the methods used and his/her preferences before reaching those difficult questions. In the case of group decisions, this way of proceeding allows to postpone (or even avoid) those elicitation questions that are more prone to cause conflict among the members of the group.

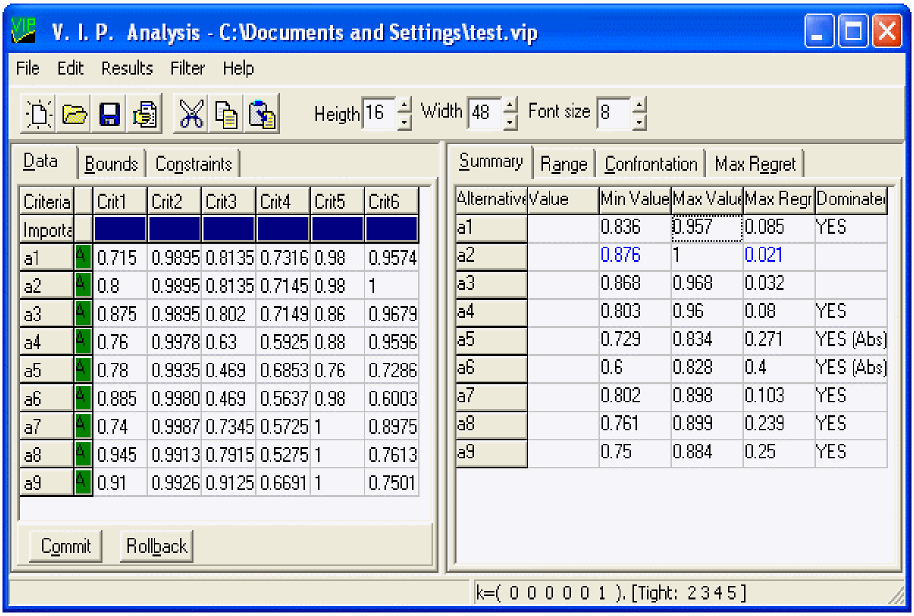
In RA phases, the robust conclusions corresponding to the current set of versions are to be discussed. It shall normally be visible that some conclusions are robust, whereas other results are very dependent on the version chosen. In group decisions, this will show where agreement exists and where disagreement is stronger. This may in turn motivate new elicitation questions or issues to be discussed, when returning to an elicitation phase.

# 4 The examples of VIP Analysis and IRIS

If the RA is adopted in the spirit of using it throughout the decision-aiding process as a means to guide an informed discussion, then RA becomes interactive. This is best achieved when there is software to aid the DM (and possibly an analyst) during the successive iterations. We next provide two examples of such software.

## 4.1 VIP Analysis (for details see Dias and Clímaco, 1999)

This software is intended to support choice decisions using additive value functions, allowing to draw robust conclusions when using different versions for the scaling weights ($k_1,...,k_n$). In elicitation phases, the DM may indicate any information that can be translated as a linear constraint, such as intervals for weights or weight ratios, parameter comparisons (e.g., $k_1 \geq k_2$), or holistic comparisons (e.g., $a_1$ is not worse than $a_2$). In RA phases, VIP Analysis uses linear programming to identify the minimum and maximum value that each alternative may achieve, as well as the minimum and maximum differences of value between each alternative and the other ones. For instance, the right half of Figure 1 depicts the minimum and maximum value for each alternative, as well as the maximum difference (max. regret) by which each alternative can lose to another, given the set of acceptable versions for the scaling weights (not shown in the figure). Dominated alternatives are those that lose to another one in all model versions; in this example, either $a_2$ or $a_3$, or both, dominate all the other alternatives. The version that is shown at the bottom of the screen (all scaling weights equal to zero except the last one) corresponds to the cell that is selected, referring to the maximum value of alternative $a_1$.




| Criteria | | Crit1 | Crit2 | Crit3 | Crit4 | Crit5 | Crit6 |
|---|---|---|---|---|---|---|---|
| Importa | | | | | | | |
| a1 | A | 0.715 | 0.9895 | 0.8135 | 0.7316 | 0.98 | 0.9574 |
| a2 | A | 0.8 | 0.9895 | 0.8135 | 0.7145 | 0.98 | 1 |
| a3 | A | 0.875 | 0.9895 | 0.802 | 0.7149 | 0.86 | 0.9679 |
| a4 | A | 0.76 | 0.9978 | 0.63 | 0.5925 | 0.88 | 0.9596 |
| a5 | A | 0.78 | 0.9935 | 0.469 | 0.6853 | 0.76 | 0.7286 |
| a6 | A | 0.885 | 0.9980 | 0.469 | 0.5637 | 0.98 | 0.6003 |
| a7 | A | 0.74 | 0.9987 | 0.7345 | 0.5725 | 1 | 0.8975 |
| a8 | A | 0.945 | 0.9913 | 0.7915 | 0.5275 | 1 | 0.7613 |
| a9 | A | 0.91 | 0.9926 | 0.9125 | 0.6691 | 1 | 0.7501 |

| Alternative | Value | Min Value | Max Value | Max Regr | Dominated |
|---|---|---|---|---|---|
| a1 | | 0.836 | 0.957 | 0.085 | YES |
| a2 | | 0.876 | 1 | 0.021 | |
| a3 | | 0.868 | 0.968 | 0.032 | |
| a4 | | 0.803 | 0.96 | 0.08 | YES |
| a5 | | 0.729 | 0.834 | 0.271 | YES (Abs) |
| a6 | | 0.6 | 0.828 | 0.4 | YES (Abs) |
| a7 | | 0.802 | 0.898 | 0.103 | YES |
| a8 | | 0.761 | 0.899 | 0.239 | YES |
| a9 | | 0.75 | 0.884 | 0.25 | YES |

Figure 1. Robust conclusions as presented in VIP Analysis

The outputs of RA indicate which alternatives are most affected by imprecision, indicating also the versions leading to the extreme results (hence inviting the DM to ponder whether such versions are acceptable or not). In a choice problematic, RA also highlights which alternatives may be discarded (dominated or quasi-dominated with respect to versions), allowing a progressive reduction of the number of alternatives.

In extending the VIP Analysis methodology to group decision aiding, Dias and Clímaco (2005) suggest how a decision panel forming a democratic decision unit may proceed to reach a final decision in a choice problem using RA. Here, the RA will aim at indicating what conclusions are robust, given the diversity of model versions originating from the group members. The idea is not to impose an aggregated model from the individual ones, but rather to reflect to each group member the consequences of his/her inputs, confronting them with analogous reflections of the group members' inputs. Each group member can study what is robust from his/her perspective and from a group perspective. Some conclusions will be robust only to a subgroup, which will be enlarged if a tolerance is considered to accept conclusions that are almost robust, bearing in mind a trade-off between this tolerance and the proportion of the group members that agree with the conclusion.

## 4.2 IRIS (for details see Dias and Mousseau, 2003)

This software is intended to support sorting decisions using ELECTRE TRI models, allowing different versions for the weights ($k_1,...,k_n$) and cutting level (λ). It implements the idea of integrating RA with an aggregation/disaggregation approach (parameter inference) proposed by Dias et al. (2002). In elicitation phases, besides linear constraints on the weights, the DM may indicate sorting examples, which should be reproduced by IRIS. In RA phases, IRIS uses linear programming to show the range of categories where each alternative may be sorted, and to infer which of the versions would satisfy the constraints with maximum slack. In Figure 2, for instance, given the set of acceptable versions for the criteria weights (not shown in the figure), the results on the right indicate the minimum and maximum category where each alternative may be sorted, as well as the sorting suggested by the inference program. When selecting the cell where the alternative "28" is assigned to category *C1* (top-left cell of the grid on the right), a version leading to that assignment appears at the bottom right of the screen. IRIS also provides some guidance when the inputs happen to be inconsistent, by proposing subsets of constraints that, if deleted (or relaxed), remove the inconsistency.

IRIS encourages the DM to interact with it through communicating sorting examples, aiming to reduce progressively the interval of categories where each alternative may be sorted. As in VIP Analysis, IRIS indicates the versions corresponding to extreme results (worst and best categories for each alternative), thus inviting the DM to ponder the acceptability of such versions.

An extension of IRIS to group decision settings has also been proposed (Damart et al., to appear). The idea is that the group may progress towards a common and unique ELECTRE TRI model while maintaining consistency (i.e., the ability to reproduce the examples with a suitable model version) along the whole process, both at the individual and at the collective levels. The interaction is based on agreements concerning sorting examples. By using an individual instance of IRIS to privately build his/her model, each DM is helped to keep a consistent set of examples. Moreover, the DM can check the impact on his/her model that results from agreeing to a sorting example being discussed by the group. At the same time, an analyst uses an instance of IRIS to maintain a consistent set of examples agreed by the group, and may use the software to answer "what-if" questions or to verify whether a "package agreement" involving several sorting decisions simultaneously is consistent or not. RA will indicate what are the robust conclusions concerning the sorting model versions of each actor, individually. For instance, if Figure 2 represents the view of a given DM, then we could assume that he would not oppose to an agreement by the group to sort alternative "28" into categories *C1*, *C2* or *C3* (see first row of the grid on the right), but he could resist an agreement to sort that alternative into category *C4*. By aggregating the robust intervals of possible categories for each alternative from all the group members, it is also possible to suggest the sorting possibilities that are more likely to generate agreement.

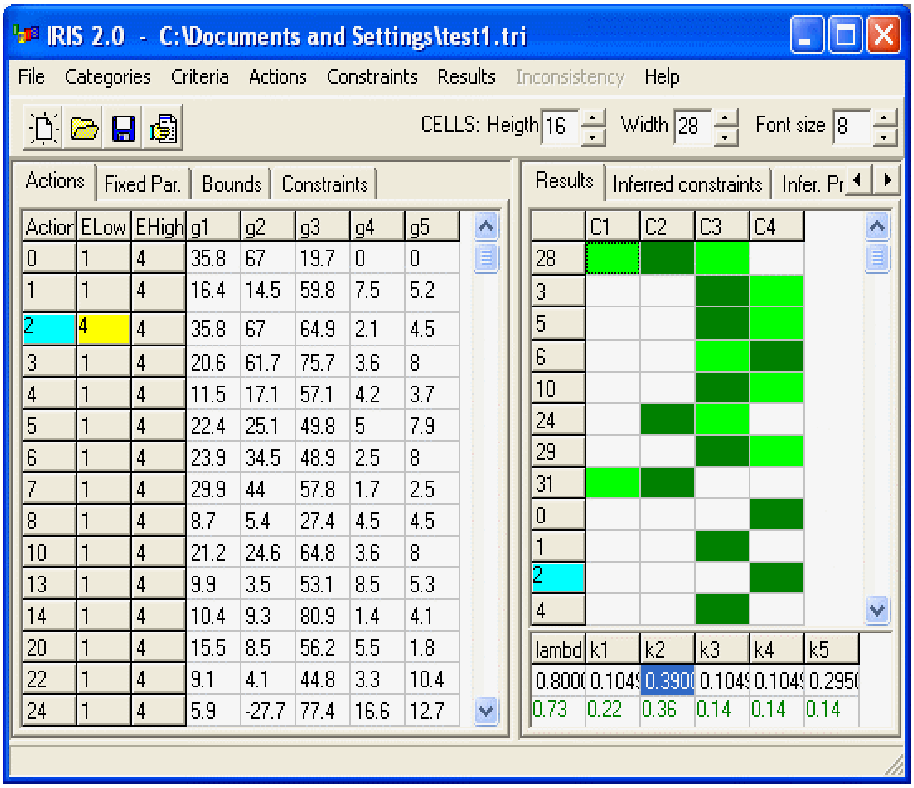



Figure 2. Robust conclusions as presented in IRIS

# 5 RA as a tool to guide a decision-aiding process

The examples of VIP Analysis and IRIS suggest some features that we believe can be useful when using RA as a tool to guide decision-aiding processes in general, namely on setting the values of preference-related parameters. This section briefly examines what this features are, to our opinion.

A first important aspect to note is the emphasis placed on the concept of robust *conclusion* (introduced in Roy and Bouyssou 1993; Roy 1998), rather than robust *solution*. For instance, a robust solution to a ranking problem will be a ranking *R* of the alternatives satisfying some property *P*, but this is the same as stating that the conclusion "*R* satisfies *P*" is robust. We may also find robust conclusions concerning part of a ranking, e.g., it is robust to conclude that alternative *a* will be better ranked than alternative *b* for all model versions. Similarly, in optimization problems we may also find robust conclusions about part of a solution (e.g., for all model versions, the shortest path between nodes *X* and *Z* passes through node *Y*). The concept of robust conclusion is hence more general than the concept of a robust solution and may allow a richer analysis.

Another important aspect to note is the emphasis placed on the idea of modeling as a constructive learning process. RA is often associated with protecting against unfavorable circumstances or unknown aspects of reality. Nevertheless, from a perspective of constructivism, which we share, not all aspects of a model are a reflex of a pre-existing (or future) reality; some aspects are constructed during the decision process (Roy 1993; Dias and Tsoukiàs 2004). This is particularly the case of many preference-related parameters of the model. When we speak about eliciting preference-related parameters such as criteria weights, we hence mean helping the DM to learn enough about the problem and his or her preferences to be able to provide some information about the values of these parameters *comfortably*. This is facilitated by allowing the DM to express information in the form of intervals, order relations, or examples of results, rather than precise numerical values.

While there are benefits in asking elicitation questions that are simple to understand and allowing the DM to answer only to the extent he/she feels comfortable, this might lead to a wide diversity of versions for the preference-related parameters that are compatible with the DM's answers. These versions might then lead to a set of robust conclusions that are considered too vague. An important feature of using RA as a tool to guide a decision-aiding process is the feedback it provides on the need to continue the elicitation effort. Elicitation and RA phases are to be reiterated until the set of conclusions that are robust is considered requisite by the DM (in the sense of (Phillips 1984)). In some cases, however, "flat maxima" (Von Winterfeldt and Edwards 1986) may allow reaching a set of surprisingly rich robust conclusions even for relatively scarce information.

Another important feature of this type of approach to decision aiding is the way robust conclusions may be used to suggest elicitation questions aimed at revising the set of model versions considered, which is modified from iteration to iteration. Robust conclusions often refer to extreme possibilities (e.g., the maximum or minimum value of an alternative or solution) and the DM may be asked whether such value is counter-intuitive (and why), or asked whether the model version that produced that extreme result is really acceptable or not (and why not). Moreover, RA will indicate what elements of the envisaged solution are stable and what elements vary more from model version to model version, helping to focus the analysis on the aspects that are seen as most relevant.

In group decisions, using RA as a tool to guide the decision process may also be helpful. Besides the extensions of VIP Analysis and IRIS outlined in the previous section, we can also mention (Lamboray 2007) as an example of this type of approach. We will consider situations where a group of DMs, such as a decision panel, genuinely wishes to reach a consensus and shares some common objectives and goals. Although the DMs may disagree on the importance of each objective, we are not considering situations where objectives are naturally opposed (e.g., buyer vs. seller on the price of a transaction). We will consider that the DMs share a set of alternatives (solutions, actions) to be evaluated, although divergence may occur in the way they evaluate the feasibility of the alternatives, or the way these alternatives perform with respect to the stated objectives. We will consider that the DMs, possibly sharing the support from a process facilitator, agree on a decision-aiding model structure to be used, although they may diverge on which parameter values should be used. In terms of *common vs. individual* ownership status of the elements (Belton and Pictet 1997), such situations may be modeled as group decisions with common decision-aid method, evaluation criteria, and list of alternatives, but individual model parameters (evaluation parameters, criteria importance parameters, etc.). Divergence hence concerns parameter values and can be taken into account through the acceptance of multiple model versions.

As for the case of a single DM, emphasis is placed on the concepts of robust conclusion and constructive learning process, and benefits are to be expected from discussing simple questions and from using robust conclusions to suggest new questions to be discussed by the group. RA will aim at indicating what conclusions are robust, given the diversity of model versions originating from the group members, providing feedback on the need to continue the group discussion.

Belton and Pictet (1997) distinguish three elementary procedures to deal with divergence among the group members. We will rephrase these elementary procedures in terms of a divergence on the model versions that should be considered:

- *Sharing* aims to obtain a common set of model versions by consensus, through a discussion of the views and the negotiation of an agreement; it addresses the differences and tries to reduce them by explicitly discussing their cause.

- *Aggregating* aims to obtain a common set of model versions (in the extreme case, a single model version) by compromise, through a vote or calculation of a representative value; it acknowledges the differences and tries to reduce them without explicitly discussing their cause.

- *Comparing* aims to reach an eventual consensus based on knowing and negotiating the independent individual results; it acknowledges the differences without necessarily trying to reduce them.

Using RA to support group decision processes in the way we advocate can fit into any of these three approaches. However, when the model versions refer to preference-related parameters, hence to parameters of a more “political” nature, we believe this type of RA fits best into a procedure of a *sharing* kind. This is particularly true if it is a goal of the group to arrive at a resolution that all members feel is jointly owned, so that all of them are committed to its successful implementation. The group therefore needs to discuss what model versions are to be envisaged and what conclusions may be drawn from those versions. From a set of shared model versions, the RA will indicate what conclusions are robust, while highlighting where divergence lies. Hence, on the one hand, RA will allow the group to control whether the set of robust conclusions is sufficient for their requisite needs; on the other hand, RA will suggest issues to be debated: results that vary to a great extent between different model versions, conclusions that are approximately robust if some model versions are discarded, whether or not to keep some model versions that lead to extreme results, etc.

Another possibility is to use RA in a procedure of the *comparing* kind. RA will then be used to reflect to each group member the consequences of his/her inputs, confronting them with analogous reflections of the group members' inputs. Each DM can compare what is robust from his/her perspective, with another DM's perspective, or with the perspective of a subset of the group, or even with all the remaining group members. Subsequently, each member can incorporate feedback from other group members (Vetschera 1991) in the revision of his/her inputs.

In the examples of VIP Analysis and IRIS for groups, particularly the latter, we find the strategies of sharing and comparing working together. Each member can perform individual RA with an individual set of model versions. Then, members can compare their robust conclusions with the conclusions of the remaining elements of the group. Finally, the group members discuss to try to reduce the divergence among the individual robust conclusions. In the case of IRIS for groups, the individual sets of model versions coexist with a common and agreed set of model versions. All this information is revised in the context of a reiterated process, possibly with the aid of a facilitator, who may propose agreements to the group members, after analyzing the answers to some "what-if" questions.

In an *aggregation* kind of procedure this type of constructive approach is less likely to be useful for preference-related parameters: first, because the possibility of manipulation becomes a concern; second, because it may hinder the sense of joint ownership of the group's resolution. Nevertheless, aggregation can still play a role in supporting and guiding the discussion among the group members. Indeed, aggregation is used in VIP Analysis and IRIS for groups to provide a richer comparison of the results from the different individuals, as well as to better characterize the agreement/divergence between them. However, aggregation is not used as a voting mechanism to settle divergence: it simply helps to inform the type of divergence that exists, helping to highlight the aspects that are more likely to be agreed by the group.

The group members are expected to discuss and resolve their divergence, rather than simply take a vote. This does not mean they have to agree completely on what model versions should or not be considered. What they have to agree on is what resolution will result from their work, and this is possible due to the many-to-one relation between model versions and robust conclusions. They may find that they strongly disagree about a parameter value yet this is irrelevant to the outcome of their work (the choice of the best alternative, for instance). Naturally, the more they manage to agree concerning the set of model versions, the richer the set of robust conclusions.

# 6 Concluding remarks

There are many distinctions that can be made regarding the approaches to deal with difficulties in setting the parameter values of decision-aiding models. In this note we have focused the distinction of RA approaches with regard to their role in a decision-aiding process that develops through time: ex-ante approaches imbed RA in a model to be optimized or to identify a subset of interesting solutions; ex-post approaches avoid *aggregating* what happens for the multiple model versions, aiming at *exploring* how robust is a result or to derive some robust conclusions about the results; a third option is to use RA as a tool to guide the process.

We have focused the role of RA as a tool to guide a decision-aiding process, prompting questions for the DM to analyze, indicating what are the results more affected by his/her answers, and showing what can be robustly concluded. As in ex-ante approaches, it intends to avoid the problem of eliciting a model version to work with. As in ex-post approaches, it follows a strategy of exploration of robust conclusions, rather than aggregation.

When the motivation for RA stems from the existence of multiple DMs, this type of approaches also seem promising as tools to guide a group decision process. In such processes, many versions may be needed to accommodate all the different views, and this set of versions can be discussed throughout an iterative process based on successive agreements. RA will not by itself reduce the divergence among the DMs; that goal is to be achieved through voluntary agreement. Nevertheless, RA will provide some guidance: it will show where disagreement is stronger, it will motivate the issues to be discussed, and it will highlight robust conclusions (agreement). VIP Analysis and IRIS were adapted to support group procedures that mix sharing and comparing of the group members' views. We feel that much more group decision procedures could be proposed using this kind of methodologies: this is an emerging research issue that deserves to be developed. In this issue, for instance, Lamboray (2007) shows that this type of interactive group processes can also be used in the context of social choice methods.

The aim of using RA as a process-guiding tool will not be to select a version, but to highlight a set of robust conclusions that is found to be requisite. Furthermore, the approach cannot be reduced to a precise recipe or path to meet the objective of the decision-aiding process. It is more like a general non-normative methodology of progressive construction (or perhaps delimitation) of a model, through the progressive reduction of the set of model versions considered, where the DM is informed by reiterated RA.

This type of approach seems particularly well-suited when the RA concerns parameters related with preferences, in that the set of versions can be reduced as a result of learning or increased effort from the DM. It may also be indicated for other parameters that can be known with higher precision but at an additional cost, e.g., data from surveys or experimental data.

# Acknowledgements

These reflections result from research work conducted with J. Clímaco, S. Damart, J. Figueira, and V. Mousseau, partially supported by FCT / FEDER projects POSI/SRI/37346/2001 and POCI/EGE/58371/2004, and program PESSOA 2004 (GRICES/EGIDE). A more condensed version of this note appeared in the 13th volume of the 3rd series of the EWG-MCDA Newsletter, edited by José Figueira (http://www.inescc.pt/~ewgmcda/Articles.html). That version is here expanded and benefited from the comments of two anonymous referees.

# References

Aloulou MA, Kalaï R, Vanderpooten D (2005), Une nouvelle approche de robustesse : α-robustesse lexicographique, *Newsletter of the EURO Working Group MCDA*, Series 3, No. 12, Fall 2005.

Belton V, Pictet J (1997), A framework for group decision using a MCDM model : sharing, aggregating or comparing individual information?, *Journal of Decision Systems* 6, 283-303.

Beer S (1966), *Decision and control: the meaning of operational research and management cybernetics*, Wiley, Chichester.

Damart S, Dias LC, Mousseau V (to appear), Supporting groups in sorting decisions: methodology and use of a multi-criteria aggregation-disaggregation DSS, *Decision Support Systems*.

Dias LC, Clíımaco JN (1999), On computing ELECTRE's credibility indices under partial information. *Journal of Multi-Criteria Decision Analysis* 8, 74–92.

Dias LC, Clímaco JN (2000), Additive aggregation with variable interdependent parameters: the VIP Analysis software, *Journal of the Operational Research Society* 51, 1070-1082.

Dias LC, JN Clímaco (2002), Exploring the consequences of imprecise information in choice problems using ELECTRE, in: Bouyssou D, E Jacquet-Lagrèze, P Perny, R Slowinski, D Vanderpooten, Ph Vincke (eds), *Aiding Decisions with Multiple Criteria - Essays in Honor of Bernard Roy*, Kluwer, 2002, 175-193.

Dias LC, JN Clímaco (2005), Dealing with imprecise information in group multicriteria decisions: a methodology and a GDSS architecture, *European Journal of Operational Research* 160, 291-307.

Dias L, Mousseau V, Figueira J, Clímaco J (2002), An aggregation/disaggregation approach to obtain robust conclusions with ELECTRE TRI, *European Journal of Operational Research*, 138, 332-348.

Dias, LC, Mousseau V (2003), IRIS: A DSS for multiple criteria sorting problems, *Journal of Multi-Criteria Decision Analysis*, 12, 285-298.

Dias, LC, Tsoukiàs A (2004), On the constructive and other approaches in decision aiding, in: C.H. Antunes, J. Figueira, J. Clímaco (eds), *Aide multicritére à la décision : Multiple criteria decision aiding*. CCDRC/INESCC/FEUC, Coimbra, pp. 13-28.

Hites R, De Smet Y, Risse N, Salazar-Neumann M, Vincke P (2003), A comparison between multicriteria and robustness frameworks, Preprint SMG/ULB 2003/16.

Janis, I. (1972). *Victims of groupthink: a psychological study of foreign-policy decisions and fiascoes*. Boston: Houghton Mifflin.

Jacquet-Lagrèze E, Siskos Y (2001), Preference disaggregation: 20 years of MCDA experience, *European Journal of Operational Research*, 130, 233-245.

Kouvelis P, Yu G (1997), *Robust discrete optimization and its applications*, Kluwer, Dordrecht.

Lamboray C (2007), Supporting the search for a group ranking with robust conclusions on prudent orders, *Annales du LAMSADE*, in this issue.

Mulvey JM, Vanderbei RJ, Zenios SA (1995), Robust optimization of large-scale systems, *Operations Research* 43, 264-281.

Phillips LD (1984), A theory of requisite decision models, *Acta Psychologica*, 56, 29-48.

Rosenhead J (2001), Robustness analysis: keeping your options open. In: J. Rosenhead and J. Mingers (eds.) *Rational Analysis for a Problematic World Revisited: problem structuring methods for complexity, uncertainty and conflict*, Wiley, Chichester, pp. 181-207.

Roy B (1989). Main sources of inaccurate determination, uncertainty and imprecision. *Mathematical and Computer Modelling*, 12, 1245-1254.

Roy B (1993), Decision Science or decision-aid science?, *European Journal of Operational Research* 66, 184-203.

Roy B (1998), A missing link in OR-DA: robustness analysis, *Foundations of Computing and Decision Sciences* 23, 141-160.

Roy B (2002). Robustesse de quoi, et vis-à-vis de quoi, mais aussi robustesse pourquoi en aide à la décision ?, *Newsletter of the EURO Working Group MCDA*, Series 3, No. 6, Fall 2002.

Roy B (2003). Commentaires sur l'article « Robustness Analysis » par J. Rosenhead, *Newsletter of the EURO Working Group MCDA*, Series 3, No. 8, Fall 2003.

Roy B (2007). La robustesse en recherche opérationnelle et aide à la décision : Une préoccupation multi facettes, *Annales du LAMSADE*, in this issue.

Roy B, Bouyssou D (1993), *Aide multicritère à la décision: méthodes et cas*, Economica, Paris.

Sevaux M, Sörensen K (2004), Robustness analysis: optimisation, *Newsletter of the EURO Working Group MCDA*, Series 3, No. 10, Fall 2004.

Vetschera R (1991), Integrating databases and preference evaluations in group decision support - A feedback-oriented approach, *Decision Support Systems*, 7, 67–77.

Vincke Ph (1999), Robust solutions and methods in decision aid, *Journal of Multi-Criteria Decision Analysis* 8, 181-187.

Von Winterfeldt D, Edwards W (1986), *Decision analysis and behavioral research*, Cambridge University Press, New York.

# Robust shortest path problems

Virginie Gabrel*, Cécile Murat*

**Résumé**

Cet article constitue un état de l'art sur les problèmes de plus courts chemins pour lesquels il existe des éléments d'incertitude et d'indétermination sur les valeurs des arcs. Deux modèles d'incertitude sont distingués : celui dit par intervalle et celui dit par scénarios. Différentes mesures et approches de la robustesse sont présentées : celles issues de la théorie de la décison, celles issues de l'analyse multicritère et celles issues de la programmation mathématique. Elles donnent lieu à plusieurs versions distinctes du problème de plus court chemin robuste dont les complexités et les résolutions sont présentées.

**Mots-clefs :** plus court chemin, optimisation robuste, incertitudes sur les données, scénario du pire cas, regret maximum

**Abstract**

This paper is a state of the art on the shortest path problems for which it exists uncertainty and inaccuracy factors on arc values. Two uncertainty models are distinguished: the so-called interval model and the discrete set of scenarios model. Different measures and approaches of robustness are presented: those coming from decision theory, those coming from multicriteria analysis and those coming from mathematical programming. Each one leads to a particular version of robust shortest path problem for which complexity and resolution are studied.

**Key words :** shortest path, robust optimization, data uncertainties, worst case scenario, maximal regret

*LAMSADE, Université Paris-Dauphine, 75775 Paris cedex 16, France. {gabrel, murat}@lamsade.dauphine.fr

# 1 Introduction

In optimization, it is used to deal with uncertain and inaccurate factors which make difficult the assignment of a single plausible value to each model parameters. Two approaches are possible : in the first one, a single nominal value is assigned to each parameter, the corresponding optimal solution is computed, then the interval in which each parameter can vary in order to preserve optimality solution is determined ; the second approach consists in taking into account in the model to optimize, the possible variations of each parameter. In mathematical programming, the first approach is known as sensibility analysis. For the second approach, stochastic optimization may be applied for some problems in which parameters' value can be described by probability laws. When it is not possible nor relevant to associate probability laws to parameters, another way amounts to assign a set of possible values to each parameter. In this context, the choice of one value in each set corresponds to a scenario. The induced robust optimization problem is to determine a single solution which is optimal for all scenarios. In general, such a solution does not exist and the problem is to determine a "relatively good" solution for all scenarios.

In most of robustness studies in combinatorial optimization, a robust solution is a solution which is acceptable in a large majority of scenarios and, which is never too bad (cf for example [23, 25]). This characterization leaves place to many possible interpretations and, thus gives place to various approaches of robustness. These approaches differ from models used to represent the uncertain and inaccurate factors of the considered decision problem, from methodology used to measure robustness, and finally from the analysis and the design of resolution methods.

In this chapter, we present various models, criteria and methods which were studied for the robust shortest path problem. In the section 2, models are proposed then comes in the section 3, the presentation of different measures of robustness, and finally, in the parts 4 and 5, is presented complexities and resolutions of various versions of the robust shortest path problem.

# 2 Models for representing uncertain, undetermined and random factors

We consider a directed valued graph $G = (X, U)$ such as $X = \{1, \ldots, n\}$ and $|U| = m$. A path $\mu$ from 1 to $n$ in $G$ is a sequence of arcs : $\mu = \{(i_0 = 1, i_1), (i_1, i_2), \ldots, (i_{k-1}, i_k = n)\}$. Within the deterministic framework, to each arc $u = (i, j) \in U$ is associated a single value $c_u = c_{ij}$. The value of $\mu$ is then given by : $c(\mu) = \sum_{(i,j)\in\mu} c_{ij}$. A traditional problem of combinatorial optimization consists in determining the shortest path, denoted by $\mu^*_G$, among the set $\mathcal{C}$ of paths from 1 to $n$ in $G$.

When elements of uncertainty and/or inaccuracy must be taken into account, it can be no more relevant, even possible, to set a single value $c_{ij}$ for each arc $(i, j)$.

In some contexts, these values are described by probability laws. The problem then becomes the stochastic shortest path problem that we will not study in this paper.

In other contexts, uncertain, inaccurate and random factors cannot be efficiently modeled by probability laws. Since the middle of the 90', two models were more particularly developed in the literature : the so-called interval model and the discrete set of scenarios model. In these two models, each arc is associated with a set of values, this set may be of infinite or finite size. The assumption is that on each arc, the real value belongs to its associated set. A scenario corresponds to the specification of a single value for each arc. We will note $S$ the set of $q$ relevant scenarios, $c_{ij}^s$ the value of arc $(i, j)$ under scenario $s$, $c^s(\mu)$ the value of path $\mu$ under scenario $s$, $G^s$ the graph valued according to the scenario $s$, $\mu^*_{G^s}$ the shortest path from 1 to $n$ in $G^s$.

In the two next sections, we present the so-called interval model and the so-called discrete set of scenarios model.

## 2.1 The interval model

In the interval model, each arc $(i, j)$ is associated to an interval, denoted $[b_{ij}, B_{ij}]$, such as $c_{ij} \in [b_{ij}, B_{ij}]$. This interval represents the set of possible values of the arc $(i, j)$. The set $S$ of scenarios is infinite and is defined as the cartesian product of the $m$ intervals $[b_u, B_u]$. In this context, a path $\mu$ can take any value belonging to the interval $[b(\mu), B(\mu)]$ with $b(\mu) = \sum_{(i,j)\in\mu} b_{ij}$ and $B(\mu) = \sum_{(i,j)\in\mu} B_{ij}$.

EXAMPLE. Let us consider the path $\mu = (1, 2, 4, 5, 7)$ in the graph $G$ of the figure 1 whose arcs are valued by intervals. Under any scenario $s$, this path has a value $c^s(\mu)$ belonging to the interval $[5, 24]$.

This model is considered for the shortest path problem in [4, 13, 14, 17, 19, 18, 28].

In addition, the interval model is applied to other combinatorial problems, for example the minimum spanning tree [16, 26], the problem of selecting $p$ elements among $n$ with minimal total weight [3], or the assignment problem [1].

## 2.2 The discrete set of scenarios model

In the discrete set of scenarios model, a finite set $S$ of scenarios is considered. Thus, given $q$ scenarios, each arc $(i, j)$ is associated to a finite set of $q$ plausible values $\{c_{ij}^1, \ldots, c_{ij}^q\}$.

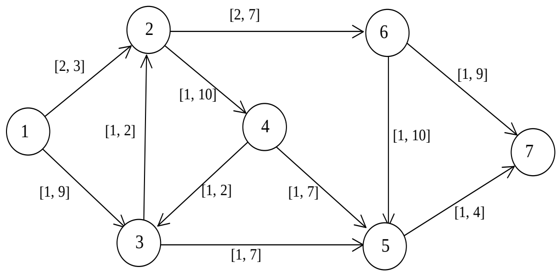



FIG. 1 – Example of interval model

Let us note that on the contrary of interval model, any combination of values $c_{ij}$ does not constitute a relevant scenario. Consequently, each path $\mu$ is associated to a $q$ dimension vector $V(\mu)$ such as $V(\mu) = (c^1(\mu), \ldots, c^q(\mu))$, where $c^s(\mu) = \sum_{(i,j)\in\mu} c^s_{ij}$ with $s \in \{1, \ldots, q\}$.

EXAMPLE. Let us consider the graph $G$ of the figure 2 whose arcs are valued according to three scenarios.

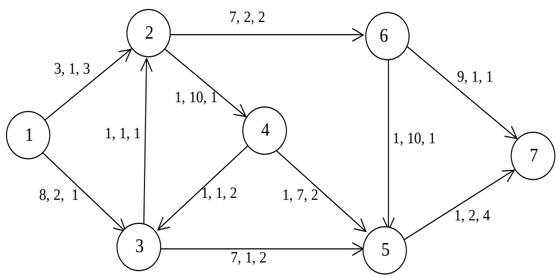



FIG. 2 – Example of scenarios model

Elementary paths from 1 to 7 and their values on the three scenarios are presented in the table 1.

This model is considered for the shortest path problem in [15, 21, 24, 27].

In addition, the discrete set of scenarios model is applied to other combinatorial problems, for example the minimum spanning tree in [15, 24], the problem of selecting $p$ elements among $n$ with minimal total weight in [3], the 1-median location problem in

| $\mu$ | $c^1(\mu)$ | $c^2(\mu)$ | $c^3(\mu)$ |
|---|---|---|---|
| $\mu_1 = (1, 2, 6, 7)$ | 19 | 4 | 6 |
| $\mu_2 = (1, 2, 6, 5, 7)$ | 12 | 15 | 10 |
| $\mu_3 = (1, 2, 4, 5, 7)$ | 6 | 20 | 10 |
| $\mu_4 = (1, 2, 4, 3, 5, 7)$ | 13 | 15 | 12 |
| $\mu_5 = (1, 3, 2, 6, 7)$ | 25 | 6 | 5 |
| $\mu_6 = (1, 3, 2, 4, 5, 7)$ | 12 | 22 | 9 |
| $\mu_7 = (1, 3, 2, 6, 5, 7)$ | 18 | 17 | 9 |
| $\mu_8 = (1, 3, 5, 7)$ | 16 | 5 | 7 |

TAB. 1 – Elementary paths from 1 to 7

[12, 15], or the assignment problem [1, 7].

A more general model is presented in [8] which includes the two previous models. In this general framework, arcs value are determined by a set of $r$ parameters. These parameters are not exactly known and two models are considered : a discrete set of parameters $r$-tuples, and a continuous subset of $R^r$ delimited by constraints (for which the interval model is a particular case).

# 3 Robustness measures

In this section, we list models to evaluate a path in term of robustness. Three approaches can be distinguished : the first one, coming from the decision theory, models the concept of robustness through a single criterion to be optimized ; the second one uses the mathematical programming to address data uncertainty ; and the third one, coming from the multicriteria analysis and the social choice theory, attempts to characterize the robust solutions by a set of conditions to be checked.

## 3.1 Classical criteria resulting from the decision theory

### 3.1.1 Worst case criterion

Given a path, the scenario to be considered is the one that gives the worst value for this path. In this context, the value of a path $\mu$, noted $v_{\text{WOR}}(\mu)$, is defined by :

$$v_{\text{WOR}}(\mu) = \max_{s \in S} c^s(\mu)$$

EXAMPLE. Let us consider the path $\mu = (1, 2, 4, 5, 7)$ of the graph in the figure 1. The worst case scenario is the one for which each arc value is equal to its upper bound. In this case, $v_{\text{WOR}}(\mu) = 24$.

The problem is to determine the path $\mu^*_{\text{WOR}}$ which minimizes $v_{\text{WOR}}$ as follows :

$$v_{\text{WOR}}(\mu^*_{\text{WOR}}) = \min_{\mu \in \mathcal{C}} v_{\text{WOR}}(\mu)$$

Under any scenario $s \in S$, $\mu^*_{\text{WOR}}$ has a value lower or equal to $v_{\text{WOR}}(\mu^*_{\text{WOR}})$. Thus, $v_{\text{WOR}}(\mu^*_{\text{WOR}})$ can be considered as an upper bound which offers an absolute guarantee. That is why in the work of Kouvelis and Yu [15], solutions which optimize this criterion are called absolute robust solutions. In decision theory, this criterion models the behavior of a decision maker averse to risk.

### 3.1.2 The maximum regret criterion

Given a scenario $s$, the choice of a path $\mu$ which is not necessarily a shortest path in $G^s$, generates a regret denoted $r^s(\mu) = c^s(\mu) - c^s(\mu^*_{G^s})$. A path $\mu$ is then evaluated by $v_{\text{REG}}(\mu)$ on the basis of the maximum regret scenario :

$$v_{\text{REG}}(\mu) = \max_{s \in S} r^s(\mu)$$

EXAMPLE. Let us consider the path $\mu = (1, 2, 4, 5, 7)$ of the graph in the figure 1 and the scenario $s^b$ in which each arc value of $U$ is set to its lower bound. Then, as the shortest path of $G^{s^b}$ is $\mu^*_{G^{s^b}} = (1, 3, 5, 7)$, we have : $r^{s^b}(\mu) = 5 - 3 = 2$. In fact, we will see in the section 4.2 that the maximum regret associated to $\mu$ is of 19, and thus : $v_{\text{REG}}(\mu) = 19$.

The optimal solution according to the maximum regret criterion will be denoted $\mu^*_{\text{REG}}$ and checks :

$$v_{\text{REG}}(\mu^*_{\text{REG}}) = \min_{\mu \in \mathcal{C}} v_{\text{REG}}(\mu)$$

EXAMPLE. Let us consider again the graph in the figure 2 for which three scenarios are taking into account. Since the shortest path in each scenario is respectively $\mu^*_{G^1} = (1, 2, 4, 5, 7)$ of value 6, $\mu^*_{G^2} = (1, 2, 6, 7)$ of value 4 and $\mu^*_{G^3} = (1, 3, 2, 6, 7)$ of value 5, we can determine the values $v_{\text{WOR}}(\mu)$ and $v_{\text{REG}}(\mu)$ for any path $\mu$, presented in the table 2.

Optimal solutions according to the worst case criterion are paths $\mu_2$ and $\mu_4$, and the optimal solution according to the maximum regret criterion is the path $\mu_8$.

In the work of Kouvelis and Yu [15], a solution which optimizes the maximum regret criterion is called a robust deviation solution. In addition, they propose another criterion of

| $\mu$ | $v_{\text{WOR}}(\mu)$ | $v_{\text{REG}}(\mu)$ |
|---|---|---|
| $\mu_1 = (1, 2, 6, 7)$ | 19 | 13 |
| $\mu_2 = (1, 2, 6, 5, 7)$ | 15 | 11 |
| $\mu_3 = (1, 2, 4, 5, 7)$ | 20 | 16 |
| $\mu_4 = (1, 2, 4, 3, 5, 7)$ | 15 | 11 |
| $\mu_5 = (1, 3, 2, 6, 7)$ | 25 | 19 |
| $\mu_6 = (1, 3, 2, 4, 5, 7)$ | 22 | 18 |
| $\mu_7 = (1, 3, 2, 6, 5, 7)$ | 18 | 13 |
| $\mu_8 = (1, 3, 5, 7)$ | 16 | 10 |

TAB. 2 – Paths value on classical criteria

robustness in term of relative deviation which consists in normalizing the regret measure as follows :

$$v_{\text{REG}'}(\mu) = \max_{s \in S} \frac{c^s(\mu) - c^s(\mu^*_{G^s})}{c^s(\mu^*_{G^s})}$$

This criterion must also be minimized.

Classical criteria coming from decision theory evaluate solutions on the basis of extreme scenarios. The major disadvantage is to focus on a single scenario, mainly the worst case, without taking into account all others scenarios. On the other hand, these criteria offer optimality guarantees. In particular, Averbakh in [2] proposes a very lighting interpretation of the maximum regret criterion. He supposes, rather naturally, that a solution is robust if it is a good solution on all the scenarios. Thus, for a scenario $s$, robust solutions must belong to the set, denoted $\mathcal{C}^s_\epsilon$, of the $\epsilon$-optimal solutions verifying :

$$c^s(\mu) - c^s(\mu^*_{G^s}) \leq \epsilon.$$

The set, denoted $\mathcal{C}^{\text{ROB}}_\epsilon$, of robust solutions is then the intersection of all $\epsilon$-optimal solutions sets, one per scenario, as follows :

$$\mathcal{C}^{\text{ROB}}_\epsilon = \bigcap_{s \in S} \mathcal{C}^s_\epsilon$$

It is obvious that if $\epsilon$ is too small, $\mathcal{C}^{\text{ROB}}_\epsilon$ will be empty, and the cardinality of $\mathcal{C}^{\text{ROB}}_\epsilon$ increases with $\epsilon$. Otherwise, if $\epsilon$ is too large, then $\mathcal{C}^{\text{ROB}}_\epsilon$ will contain solutions with bad evaluations on some scenarios. Consequently, it is particularly interesting to determine the smallest value of $\epsilon$ such as $\mathcal{C}^{\text{ROB}}_\epsilon$ contains at least one solution. Averbakh shows that this value is not other than $v_{\text{REG}}(\mu^*_{\text{REG}})$. Indeed, solutions of $\mathcal{C}^{\text{ROB}}_\epsilon$ are such that :

$$\begin{aligned} & \forall \mu \in \mathcal{C}^{\text{ROB}}_\epsilon, \quad \forall s \in S, \; c^s(\mu) - c^s(\mu^*_{G^s}) \leq \epsilon \\ \Rightarrow \; & \forall \mu \in \mathcal{C}^{\text{ROB}}_\epsilon, \quad \max_{s \in S} \left(c^s(\mu) - c^s(\mu^*_{G^s})\right) = v_{\text{REG}}(\mu) \leq \epsilon \end{aligned}$$

But, since $\forall \mu \in \mathcal{C}_{\epsilon}^{\text{ROB}}$, $v_{\text{REG}}(\mu) \geq v_{\text{REG}}(\mu_{\text{REG}}^{*})$, if $\epsilon < v_{\text{REG}}(\mu_{\text{REG}}^{*})$ then $\mathcal{C}_{\epsilon}^{\text{ROB}}$ is empty, and otherwise, $\mathcal{C}_{\epsilon}^{\text{ROB}}$ contains at least $\mu_{\text{REG}}^{*}$. Consequently, $v_{\text{REG}}(\mu_{\text{REG}}^{*})$ is the smallest $\epsilon$ such as it exists at least one solution which is $\epsilon$-optimal for all scenarios. This interpretation may be adapted to the relative deviation criterion by modifying the definition of $\epsilon$-optimal solution as follows : for a scenario $s$, $\mu$ is $\epsilon$-optimal if and only if $\frac{c^s(\mu)-c^s(\mu_{Gs}^{*})}{c^s(\mu_{Gs}^{*})} \leq \epsilon$.

## 3.2 Methodology coming from mathematical programming

The shortest path problem can be represented by an integer linear program of the form :

$$(\mathcal{P}) \begin{cases} \min & \sum_{(i,j)\in U} c_{ij} y_{ij} \\ s.c & y \in Y \end{cases}$$

In studies dealing with robustness in mathematical programming, we can distinguish those in which data uncertainty affects only the elements of the constraint matrix, and those in which data uncertainty concerns only objective function coefficients. In this context, a very interesting approach is introduced by Bertsimas and Sim (in [5] and [6]). They consider the following interval model for objective function coefficients : $c_{ij} \in [\overline{c}_{ij}, \overline{c}_{ij} + \widehat{c}_{ij}]$ where $\overline{c}_{ij}$ is the nominal value for $(i,j)$ and $\widehat{c}_{ij} \geq 0$ represents the deviation from the nominal coefficient $\overline{c}_{ij}$. They start from the quite natural idea that the worst case will not happen simultaneously for all coefficients. In [5], authors "stipulate that nature will be restricted in its behavior, in that only a subset of coefficients will change in order to adversely affect the solution".

So, they introduce a parameter $\Gamma$ which represents the maximum number of coefficients that can deviate from their nominal value : $\Gamma = 0$ means that none coefficient will vary, while $\Gamma = m$ means that all coefficients will vary in the worst case sense. So, $\Gamma$ is interpreted as a level of robustness.

The robust version of the shortest path problem becomes :

$$(\mathcal{P}_{\text{ROB}}) \begin{cases} \min & \left( \sum_{(i,j)\in U} \overline{c}_{ij} y_{ij} + \max_{\{R \mid R\subseteq U, |R| \leq \Gamma\}} \sum_{(ij)\in R} \widehat{c}_{ij} y_{ij} \right) \\ s.c & y \in Y \end{cases}$$

## 3.3 Methodology coming from multicriteria analysis

In multicriteria analysis [22], a decision problem is commonly defined using a set of solutions, a discrete set of criteria and an aggregation model of these criteria. A solution is thus described by its evaluation vector containing the solution value on each criterion. Considering a scenario as a criterion, a part of the results obtained in multicriteria analysis can be used for the robustness analysis (cf [11]). In this context, the robustness analysis is based upon the choice of an evaluation vector associated with each solution, and the definition of an aggregation model of these evaluation vectors.

### 3.3.1 Path evaluation vector

When the set of scenarios is discrete, it is natural to consider the vector $V(\mu)$, representing all possible values of $\mu$ on various scenarios, as an evaluation vector (cf [8] and [11]).

More recent work ([12] and [21]) applies the anonymous principle which consists in admitting that the vector $V(\mu) = (c^1(\mu), \ldots, c^q(\mu))$ is equivalent to the vector $V(\mu) = (c^{\sigma(1)}(\mu), \ldots, c^{\sigma(q)}(\mu))$ for any permutation $\sigma(i)$ of the $q$ scenarios. According to this principle, a path $\mu$ can be evaluated by a vector obtained by ordering the values $c^i(\mu)$ in a decreasing order. This vector is called disutility vector in [12], and is denoted $D(\mu) = (d^1(\mu), \ldots, d^q(\mu))$, where $d^1(\mu) \geq \ldots \geq d^q(\mu)$ with $d^i(\mu)$ representing the value of order $i$ of the path $\mu$.

In some other studies on robustness, another so-called equity principle is proposed. According to this principle, a solution whose evaluations are well distributed around the average on the various scenarios is always preferred to a more unbalanced solution. For that, one evaluates a solution on the basis of the following vector, called in [21] the generalized Lorenz vector : $L(\mu) = (l^1(\mu), \ldots, l^q(\mu))$ with $l^j(\mu) = \sum_{i=1}^{j} d^i(\mu)$.

EXAMPLE. For the graph in figure 2, desutility and generalized Lorenz vectors are presented for each path in the table 3.

| | Desutility vectors | | | Lorenz vectors | | |
|---|---|---|---|---|---|---|
| $\mu$ | $d^1(\mu)$ | $d^2(\mu)$ | $d^3(\mu)$ | $l^1(\mu)$ | $l^2(\mu)$ | $l^3(\mu)$ |
| $\mu_1 = (1,2,6,7)$ | 19 | 6 | 4 | 19 | 25 | 29 |
| $\mu_2 = (1,2,6,5,7)$ | 15 | 12 | 10 | 15 | 27 | 37 |
| $\mu_3 = (1,2,4,5,7)$ | 20 | 10 | 6 | 20 | 30 | 36 |
| $\mu_4 = (1,2,4,3,5,7)$ | 15 | 13 | 12 | 15 | 28 | 40 |
| $\mu_5 = (1,3,2,6,7)$ | 25 | 6 | 5 | 25 | 31 | 36 |
| $\mu_6 = (1,3,2,4,5,7)$ | 22 | 12 | 9 | 22 | 34 | 43 |
| $\mu_7 = (1,3,2,6,5,7)$ | 18 | 17 | 9 | 18 | 35 | 44 |
| $\mu_8 = (1,3,5,7)$ | 16 | 7 | 5 | 16 | 23 | 28 |

TAB. 3 – Evaluation vectors for paths from 1 to 7

### 3.3.2 Aggregation models for robustness

In multicriteria analysis, the dominance relation is defined as follows.

DEFINITION. Given a family $F = (f^1, \ldots, f^q)$ of $q$ criteria, a path $\mu$ dominates a path $\mu'$ if and only if $f^i(\mu) \leq f^i(\mu')$ for $i = 1, \ldots, q$, with at least one strict inequality. The path $\mu'$ is then dominated by $\mu$, that one notes $\mu \Delta_F \mu'$.

DEFINITION. A path $\mu$ is non dominated if and only if there is not another path $\mu' \in \mathcal{C}$ such as $\mu' \Delta_F \mu$.

In the interval model, it is not possible to associate to each path a finite number of evaluation vectors because the number of scenarios is infinite. However, dominance relation remains defined in this model as follows

DEFINITION. A path $\mu$ dominates a path $\mu'$ if and only if $c^s(\mu) \leq c^s(\mu')$ for all $s \in S$, with at least one strict inequality. The path $\mu'$ is then dominated by $\mu$ that one notes $\mu \Delta \mu'$.

Whatever the evaluation vector, robust solutions naturally belong to the non-dominated solutions set. Thus, the problem can be first to determine all non-dominated solutions (see for example [8]).

EXAMPLE. For the graph in figure 2, and regarding to the evaluation vector $V(\mu)$ (in table 1), all solutions are non-dominated excepted $\mu_4$ which is dominated by $\mu_2$ ; regarding to the desutility vector (in table 3), non-dominated solutions are $\mu_1$, $\mu_2$, $\mu_8$ ; and regarding to the Lorenz vector, non-dominated solutions are $\mu_2$ and $\mu_8$. Let us remark that the path $\mu_4$ which is optimal for the worst case criterion is a dominated path.

The main difficulty comes from the possible huge number of non-dominated paths. Hansen in [10] has defined a bicriteria graph for which the number of non-dominated paths is an exponential function of the graph order. Perny and Spanjaard showed in [21] that the non-dominated solutions set according to Lorenz evaluation vectors is a subset of the non-dominated solutions set according to $V(\mu)$ vectors. Thus, in the graph proposed by Hansen, only two paths remain non-dominated according to Lorenz evaluation vectors. However, in [21], a graph with an exponential number of non-dominated paths according to Lorenz evaluation vectors is proposed.

Consequently, the determination of all non-dominated solutions is not possible in general case. Other procedures must be defined for aggregating evaluation vectors in order to provide one or a small size set of robust solutions. Let us mention an alternative approach based on a lexicographic procedure. This approach is applied in [12] on the disutility vector for the 1-median location problem.

These three approaches are quiet different. Classical criteria coming from decision theory only take into account extreme scenario. The methodology introduced by Bertsimas and Sim represents a way to qualify worst case analysis by considering that only a subset of coefficients will vary in the bad sense. In the approach coming from the decision theory, a solution is robust when it optimizes a particular criterion. In the approach coming from multicriteria analysis, the concept of robustness is not characterized anymore by optimality on a single criterion, but by a list of principles and conditions that leads to determine a set of robust solutions (this set being able to be empty !).

# 4 Complexity and resolution of robust shortest path problems in the interval model

## 4.1 With the worst case criterion

This version of the robust shortest path problem, noted ROBINTWOR, is studied in [13]. Let us recall that, in this version, a path $\mu$ has a value $v_{\text{WOR}}(\mu) = \max_{s \in S} c^s(\mu)$ and the problem is to determine the optimal path that minimizes this value on all paths. Given a path $\mu$ from $1$ to $n$, the scenario which maximizes $c^s(\mu)$ is the one for which each arc $u \in U$ has the higher possible value, that is to say $B_u$. Since this worst case scenario, noted $s^B$, is unique, it is enough to consider the graph evaluated by this worst case scenario, and to determine the shortest path which exactly corresponds to $\mu^*_{\text{WOR}}$ since $v_{\text{WOR}}(\mu^*_{\text{WOR}}) = min_{\mu \in \mathcal{C}} B(\mu)$. Consequently,

THEOREM. [13] The ROBINTWOR problem is polynomial.

EXAMPLE. In the graph of the figure 1, the optimal path according to the worst case criterion is $\mu^*_{\text{WOR}} = (1, 2, 6, 7) = \mu_1$ of value $19$.

## 4.2 With the maximum regret criterion

This version of the robust shortest path problem, denoted ROBINTREG, is studied in [4, 13, 17, 18, 19, 28]. Most results are based on the following property, established by Karaşan, Pinar and Yaman. Let us first recall that, in this problem version, a path $\mu$ has the value $v_{\text{REG}}(\mu) = \max_{s \in S} r^s(\mu) = \max_{s \in S}(c^s(\mu) - c^s(\mu^*_{G^s}))$.

**Propriété 1** *[13] Given a path $\mu$ from $1$ to $n$, the scenario, denoted $s(\mu)$, which maximizes $r^s(\mu)$ is the one for which each arc $(i, j)$ belonging to $\mu$ has a value equals to its* *higher possible value, in other words* $c_{ij}^{s(\mu)} = B_{ij}$*, and each arc* $(k, h)$ *not belonging to* $\mu$ *has a value equals to its lower bound, in other words* $c_{kh}^{s(\mu)} = b_{kh}$.

The scenario $s(\mu)$ is called the scenario induced by $\mu$. The problem to determine the robust path, according to the maximum regret criterion, is to find $\mu^*_{\text{REG}}$ such as :

$$v_{\text{REG}}(\mu^*_{\text{REG}}) = \min_{\mu \in \mathcal{C}} (c^{s(\mu)}(\mu) - c^{s(\mu)}(\mu^*_{G^{s(\mu)}}))$$

EXAMPLE. For the graph of the figure 1, the table 4 presents the maximum regret value of each path.

| $\mu$ | $c^{s(\mu)}(\mu)$ | $c^{s(\mu)}(\mu^*_{G^{s(\mu)}})$ | $v_{\text{REG}}(\mu)$ |
|---|---|---|---|
| $\mu_1 = (1, 2, 6, 7)$ | 19 | 3 | 16 |
| $\mu_2 = (1, 2, 6, 5, 7)$ | 24 | 6 | 18 |
| $\mu_3 = (1, 2, 4, 5, 7)$ | 24 | 5 | 19 |
| $\mu_4 = (1, 2, 4, 3, 5, 7)$ | 26 | 5 | 21 |
| $\mu_5 = (1, 3, 2, 6, 7)$ | 27 | 5 | 22 |
| $\mu_6 = (1, 3, 2, 4, 5, 7)$ | 32 | 5 | 27 |
| $\mu_7 = (1, 3, 2, 6, 5, 7)$ | 32 | 8 | 24 |
| $\mu_8 = (1, 3, 5, 7)$ | 20 | 5 | 15 |

TAB. 4 – Maximum regret value of each path from 1 to 7

The path $\mu_8$ is optimal for the maximum regret criterion ($\mu_8$ is also the optimal path for the best case criterion).

Unfortunately, the difficulty comes from the fact that the number of elementary paths from 1 to $n$ may be exponential.

### 4.2.1 Complexity

This problem was shown to be NP-hard for directed graphs by Zieliński in [28] with a reduction to a particular version of the partition problem known as being NP-hard (see on this subject [9]). In parallel, Averbakh and Lebedev established in [4] that ROBINTREG is strongly NP-hard for non-directed graphs, by making a reduction to the Hamiltonian path problem, denoted HAMIL. They notice that their result can be easily extended for the case of acyclic directed graphs with a layered structure. In the following, we present this extension. Let us first recall that a graph has a layered structure if it is possible to partition the vertices set $X$ into disjoint subsets $X_1, \ldots, X_g$, called blocks, such as each arc goes from a vertex of a block $X_j$ to a vertex of the next block $X_{j+1}$.

THEOREM. [4] The problem ROBINTREG is NP-hard even for the acyclic directed graphs with a layered structure.

To prove this theorem, we have to introduce the decision version of the problem ROBINTREG which is defined by :

**Instance :** an acyclic directed graph $G = (X, U)$ with a layered structure, having two particular vertices $1$ and $n$, whose arcs $u \in U$ are valued by intervals $[b_u, B_u]$ and an integer $M$,

**Question :** does there exist a path from $1$ to $n$ in $G$ with a maximum regret value lower or equal to $M$ ?

The HAMIL problem is defined as follows :

**Instance :** a non-directed connected graph $F = (V, E)$.

**Question :** does there exist in $F$ an Hamiltonian chain, that is to say a chain containing each vertex only once ?

An instance $F = (V, E)$ of this problem, with $|V| = p$, can be polynomially reduced to an instance of the decision problem ROBINTREG in the following way. Vertices of $G$ are vertices $1$, $n$ and $2p$ copies of the set $V$. The copy numbered $i$, denoted $V_i$, is the block $i$ of $G$. For any vertex $v \in V$, we note $v_i$ its copy in the block $i$, for $i = 1, \ldots, 2p$. Thus, we have $X = (1, V_1, \ldots, V_{2p}, n)$, with $|X| = 2 + 2p^2$. The set $U$ of $G$ is defined as follows :

- arcs $(1, v)$, for any vertex $v \in V_1$,
- arcs $(v, n)$, for any vertex $v \in V_{2p}$,
- arcs $(v_i, v_{i+1})$, for any vertex $v \in V$ with $i = 1, \ldots 2p - 1$, these arcs will be called horizontal arcs,
- arcs $(v_{2i}, w_{2i+1})$ and $(w_{2i}, v_{2i+1})$, for all $i = 1, \ldots p - 1$ and for all arcs $(v, w)$ belonging to $E$.

The set of horizontal arcs is noted $H$. Any arc of $U$ which is not horizontal is called diagonal. The set of diagonal arcs is denoted $D$, among which we distinguish the subset $\tilde{D}$ containing arcs of $D$ not linked to $1$ nor $n$. Intervals associated to horizontal arcs are $[0, 1]$, and those associated to diagonal arcs are $[1, 1]$.

In the figure 3, an example of graph $F$ with $3$ vertices is represented on the left and the graph $G$ corresponding to the reduction is given on its right (in which intervals $[1, 1]$ are replacing by the value 1). Let us observe that the vertices of the even blocks $V_{2i}$, $i = 1, \ldots, p$ can be reached by some vertices of the block $V_{2i-1}$ only with horizontal arcs. For any vertex $v \in V$, the path $(1, v_1, \ldots, v_{2p}, n)$ is called a $v$-tunnel.

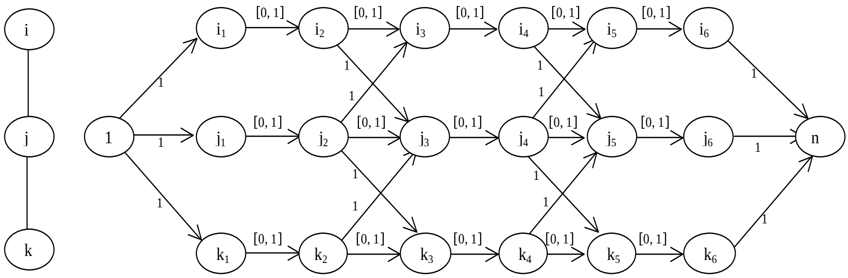



FIG. 3 – Example of the reduction of Hamiltonian path problem to ROBINTREG problem

It should be shown that an instance of HAMIL problem has a positive answer if and only if the corresponding instance of ROBINTREG problem has a positive answer for $M = 2p - 2$.

For a path $\mu$ from $1$ to $n$, we denote $|\mu|$ its number of arcs and $\overline{\mu}$ its vertices belonging to $V_1$,..., $V_{2p}$. By definition of scenario induced by $\mu$ and according to upper bounds on arcs of the graph $G$, we have : $c^{s(\mu)}(\mu) = |\mu| = 2p + 1$. Moreover under scenario $s(\mu)$, any arc of $D$ and any arc of $H$ which belongs to $\mu$ has a unit cost, whereas the cost of any arc of $H$ which does not belong to $\mu$ is equal to 0. Consequently, the cost of a path $\lambda$ from $1$ to $n$ in $s(\mu)$ is : $c^{s(\mu)}(\lambda) = |D \cap \lambda| + |H \cap \lambda \cap \mu| = 2 + |\tilde{D} \cap \lambda| + |H \cap \lambda \cap \mu|$. With $f(\mu, \lambda) = c^{s(\mu)}(\mu) - c^{s(\mu)}(\lambda)$, we have :

$$f(\mu, \lambda) = 2p - 1 - |\tilde{D} \cap \lambda| - |H \cap \lambda \cap \mu| \tag{1}$$

Let us suppose that there exists an Hamiltonian chain in $F$ denoted $(v^1, \ldots, v^p)$. Then, the sequence $(1, v_1^1, v_2^1, v_3^2, v_4^2, \ldots, v_{2p-1}^p, v_{2p}^p, n)$ is a path $\mu$ from $1$ to $n$ in $G$. Since each tunnel $T$ does not have any arc in $\tilde{D}$ and exactly one horizontal arc in common with $\mu$, we have $f(\mu, T) = 2p - 2$. Moreover, since any path $\lambda$ from $1$ to $n$ which is not a tunnel has at least one arc in $\tilde{D}$, we have : $f(\mu, \lambda) \leq 2p - 2$. Consequently,

$$v_{\text{REG}}(\mu) = \max_{\lambda \in \mathcal{C}} f(\mu, \lambda) = 2p - 2$$

Let us now consider an optimal path $\mu$ from $1$ to $n$ whose maximum regret is lower or equal to $2p - 2$. $T$ is a tunnel such that $k = |H \cap T \cap \mu| = |\overline{T} \cap \overline{\mu}|$ is minimum. Since $\mu$ is optimal with a value lower or equal to $2p - 2$, and according to the equation 1, we have : $f(\mu, T) = 2p - 1 - k \leq 2p - 2$. This implies $k \geq 1$. Thus each tunnel intersects $\overline{\mu}$ with at least $k$ different horizontal arcs, and one needs at least $p - 1$ diagonal arcs in $\overline{\mu}$ to connect together these $k$ arcs. So, we have : $|\overline{\mu}| = 2p - 1 \geq kp + p - 1$ implying $k \leq 1$. Consequently, $k = 1$, $f(\mu, T) = 2p - 2$ and $v_{\text{REG}}(\mu) = 2p - 2$ with exactly one horizontal arc of the form $(v_j, v_{j+1})$ $\forall v \in V$ and exactly one diagonal arc of the form $(v_{2i}, w_{2i+1})$ $\forall I = 1, \ldots, p - 1$ in $\mu$. Finally, we obtain an Hamiltonian chain in $F$ from the diagonal arcs of $\overline{\mu}$.

### 4.2.2 Resolution

Several algorithms for exactly solving ROBINTREG have been proposed [13, 17, 18, 19]. In these studies, ROBINTREG is written as an integer linear program with variables $y_{ij}$ which represent a path $\mu$ as follows :

$$y_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mu \\ 0 & \text{otherwise} \end{cases}$$

The integer linear program, denoted $\mathcal{P}^0$, is :

$$\mathcal{P}^0 \left\{ \begin{array}{lll} \min & \max_{s \in S} \left( \sum_{(i,j) \in U} c_{ij}^s y_{ij} - x^s \right) & \\ s.c. & \sum_{(i,j) \in U} y_{ij} - \sum_{(k,i) \in U} y_{ki} = \begin{cases} 1 & \text{if } i = 1 \\ -1 & \text{if } i = n \\ 0 & \text{otherwise} \end{cases} & \forall i \in X \\ & y_{ij} \in \{0,1\} \ \forall (i,j) \in U & \end{array} \right.$$

where $x^s$ is the length of the shortest path from 1 to $n$ in $G^s$.

According to the property 1 and given a path $\mu$ represented by $y$, the scenario which carries out $\max_{s \in S} \sum_{(i,j) \in U} c_{ij}^s y_{ij} - x^s$ is the induced scenario $s(\mu)$. Under this scenario, the length of any arc $(i, j)$ can be written in function of $y_{ij}$ as follows : $b_{ij} + (B_{ij} - b_{ij}) y_{ij}$. Thus, by introducing variables $x_i$ which represent the length of the shortest path from 1 to $i$ under scenario $s(\mu)$, Karaşan et al. in [13] propose a new formulation of ROBINTREG, noted $\mathcal{P}^1$.

$$\mathcal{P}^1 \left\{ \begin{array}{llll} \min & \sum_{(i,j) \in U} B_{ij} y_{ij} - x_n & & \\ s.c & \sum_{(i,j) \in U} y_{ij} - \sum_{(k,i) \in U} y_{ki} = \begin{cases} 1 & \text{if } i = 1 \\ -1 & \text{if } i = n \\ 0 & \text{otherwise} \end{cases} & \forall i \in X & (1) \\ & x_j \leq x_i + b_{ij} + (B_{ij} - b_{ij}) y_{ij} & \forall (i,j) \in U & (2) \\ & x_1 = 0 & & (3) \\ & y_{ij} \in \{0,1\} & \forall (i,j) \in U & \\ & x_i \geq 0 & \forall i \in X & \end{array} \right.$$

$\mathcal{P}^1$ is thus a mixed linear program. Karaşan et al. in [13] solve it using a linear programming solver. However, when $G$ is of big size, $\mathcal{P}^1$ cannot be solved exactly. The difficulty comes from the constraints $(2)$ which, if deleated, would reduce $\mathcal{P}^1$ to a simple shortest path problem. Also, in [13], authors propose to apply a preprocessing procedure which removes some constraints $(2)$, those associated with some arcs that cannot belong to the optimal path. By reducing the size of $\mathcal{P}^1$, this procedure makes possible to increase the size of exactly solved problems.

More recently, Montemanni et al. proposed two algorithms to solve exactly the problem : a dedicated algorithm presented in [17], and a branch and bound algorithm presented in [19]. In [18], authors propose to apply a Benders decomposition scheme on $\mathcal{P}^1$. A great number of experiments is carried out to compare these various approaches. It seems that the algorithm resulting from the Benders decomposition is most powerful to solve most of the considered problems.

### 4.3 With the multicriteria analysis methodology

In the interval model, a path $\mu$ dominates a path $\mu'$, denoted by $\mu\Delta\mu'$, either if $\delta^* < 0$ with $\delta^* = \max_{s\in S} c^s(\mu) - c^s(\mu')$, or $[\delta^* = 0$ and $\exists s \in S : c^s(\mu) < c^s(\mu')]$. The complexity of this dominance test comes from the infinite set of scenarios to be considered. Dias and Climaco studied this problem in [8] for a more general model (for which the interval model is a particular case). They introduced binary relations to compare path evaluation vectors on the basis of two extreme scenarios only, the best case and the worst case scenarios (already introduced in section 4.1). In the best case scenario, denoted $s^b$, the value of each arc $u \in U$ is equal to the smallest cost, that is to say $c_u^{s^b} = b_u$. They defined the following binary relations :

**Relation 1 :** $\mu \ll \mu'$ if and only if $c^{s^B}(\mu) < c^{s^b}(\mu')$

**Relation 2 :** $\mu < \mu'$ if and only if $[c^{s^b}(\mu) = c^{s^b}(\mu')$ and $c^{s^B}(\mu) < c^{s^B}(\mu')]$, or $[c^{s^b}(\mu) < c^{s^b}(\mu')$ and $c^{s^B}(\mu) = c^{s^B}(\mu')]$

**Relation 3 :** $\mu <\approx \mu'$ if and only if $c^{s^b}(\mu) \leq c^{s^b}(\mu')$ and $c^{s^B}(\mu) \leq c^{s^B}(\mu')$

On the basis of these three relations, they established the following results in the comparison of two paths $\mu$ and $\mu'$ of $\mathcal{C}$ :

- $\mu \ll \mu' \Rightarrow \mu\Delta\mu'$.
- $c^{s^b}(\mu') < c^{s^b}(\mu) \Rightarrow \mu$ cannot dominate $\mu'$.
- $c^{s^B}(\mu') < c^{s^B}(\mu) \Rightarrow \mu$ cannot dominate $\mu'$.
- $\mu\Delta\mu' \Rightarrow \mu <\approx \mu'$, with the corollary : if $\mu$ is not $<\approx$-dominated then $\mu$ is not dominated.

From these results, they proposed an algorithm, based on the enumeration of $k$ shortest paths, to determine the set of non-dominated paths. In this algorithm, paths are enumerated in ascending order of their value on the scenario $s^b$, then $<\approx$-dominated and $\ll$-dominated paths are removed, and finally, the complete dominance test is carried out only on remaining paths in order to eliminate the dominated ones.

In the specific case of interval model, the complete dominance test has not to be carried out. The property below allows to conclude by considering only the extreme best and worst case scenarios. Let us first denote by $\mu \setminus \mu'$ the set of arcs that belong to $\mu$ and not to $\mu'$.

**Propriété 2** *$\mu$ dominate $\mu'$ if and only if $c^{s^B}(\mu \setminus \mu') \leq c^{s^b}(\mu' \setminus \mu)$.*

PROOF. Let us show that $c^{s^B}(\mu \setminus \mu') \leq c^{s^b}(\mu' \setminus \mu) \Rightarrow \mu\Delta\mu'$.
$c^{s^B}(\mu \setminus \mu') \geq c^{s}(\mu \setminus \mu'), \forall s \in S$ and $c^{s^b}(\mu' \setminus \mu) \leq c^{s}(\mu' \setminus \mu), \forall s \in S$. Consequently, $c^{s^B}(\mu \setminus \mu') \leq c^{s^b}(\mu' \setminus \mu)$ implies :

$$\begin{aligned} &\Rightarrow \quad c^s(\mu \setminus \mu') \leq c^s(\mu' \setminus \mu), \forall s \in S \\ &\Leftrightarrow \quad c^s(\mu \setminus \mu') + \sum_{u \in \mu \cap \mu'} c_u^s \leq c^s(\mu' \setminus \mu) + \sum_{u \in \mu' \cap \mu} c_u^s, \forall s \in S \\ &\Leftrightarrow \quad c^s(\mu) \leq c^s(\mu'), \forall s \in S \end{aligned}$$

Let us show now that $\mu\Delta\mu' \Rightarrow c^{s^B}(\mu \setminus \mu') \leq c^{s^b}(\mu' \setminus \mu)$.
Under scenario $s(\mu)$, $c^{s(\mu)}(\mu) = c^{s^B}(\mu)$ and $c^{s(\mu)}(\mu') = c^{s^b}(\mu') + \sum_{u \in \mu' \cap \mu}(B_u - b_u)$. But, with the dominance property of $\mu$ for the specific scenario $s(\mu)$, we obtain :

$$\begin{aligned} c^{s^B}(\mu) \quad &\leq \quad c^{s^b}(\mu') + \sum_{u \in \mu' \cap \mu} (B_u - b_u) \\ \Leftrightarrow c^{s^B}(\mu \setminus \mu') \quad &\leq \quad c^{s^b}(\mu' \setminus \mu) \end{aligned}$$

REMARK. If $\mu$ and $\mu'$ have not any joint arc, then the property 2 becomes : $\mu\Delta\mu' \Leftrightarrow c^{s^B}(\mu) \leq c^{s^b}(\mu')$.

EXAMPLE. In the graph represented in the figure 4, $\mu = (a, b, d, e)$ dominates $\mu' = (a, c, d, e)$. And, it is checked that $c^{s^B}(\mu \setminus \mu') = 5 < c^{s^b}(\mu' \setminus \mu) = 6$ with $c^{s^B}(\mu) = 11 > c^{s^b}(\mu') = 10$.

From the property 2, it is possible to propose an algorithm, similar to the one proposed by Dias and Climaco in [8], based on the enumeration of the $k$ shortest paths. Ones have to enumerate paths in ascending order of their value on the best case scenario and, in the event of equality, by ascending order of their value on the worst case scenario. Consequently, the ith enumerated path, denoted $\mu_i$, cannot be dominated by $\mu_k$ if $i < k$. On the other hand, $\mu_i$ dominates $\mu_k$ if and only if $c^{s^B}(\mu_i \setminus \mu_k) \leq c^{s^b}(\mu_k \setminus \mu_i)$. This test is the single test to be carried out for each enumerated path.

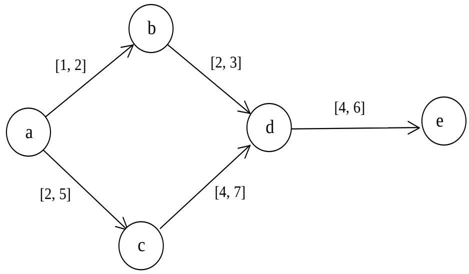



FIG. 4 – Dominance test for graph with intervals

## 4.4 With the mathematical programming methodology

In [5], authors show that the problem $\mathcal{P}_{\text{ROB}}$ presented in section 3.2 can easily be solved by solving at most $m+1$ shortest path problems. In fact, the optimal solution value of $\mathcal{P}_{\text{ROB}}$, denoted $Z^\star$ is given by :

$$Z^\star = \min_{k=1,\dots m} Z^k$$

where

$$Z^k = \Gamma\widehat{c}_k + \min_{y \in Y}\left(\sum_{(i,j)\in U} \overline{c}_{ij} y_{ij} + \sum_{h=1}^{k} (\widehat{c}_h - \widehat{c}_k) y_h\right)$$

where it is assumed that the indices of the arcs are ordered such that $\widehat{c}_1 \geq \widehat{c}_2 \geq \dots \widehat{c}_m$.

Consequently, in this approach, the robust version presents the same complexity as the nominal one. Thus, in the case of the shortest path problem, the robust version can be polynomially solved which makes this approach very attractive.

# 5 Complexity and resolution of robust shortest path problems in the discrete set of scenarios model

## 5.1 With the worst case criterion

### 5.1.1 Complexity

This version of the robust path problem, denoted ROBDISWOR, is studied in [15, 27]. Yu and Yang show that this problem is weakly NP-difficult if the number of scenarios is bounded by a constant and strongly NP-difficult otherwise. Indeed, they show in [27] that the 2-partition problem

**Instance :** a set $I$ of $m$ elements and a size $a_i \in \mathbb{Z}_+$ for each $i$ in $I$.

**Question :** Does there exist a sub-set $I' \subseteq I$ such as $\sum_{i \in I'} a_i = \sum_{i \in I \setminus I'} a_i$ ?

is reduced to ROBDISWOR problem defined as follows :

**Instance :** a connected graph $G = (X, U)$ with 2 scenarios.

**Question :** Does there exist a path $\mu$ from 1 to $n$ such as $v_{\text{WOR}}(\mu) = v$ ?

For that, we only have to consider the graph $G = (X, U)$ reduced to the 2-partition problem (represented in the figure 5) without directed cycle valued by two scenarios. This graph presents $2m + 2$ levels with $X = X_0 \cup X_1 \cup \ldots \cup X_{2m+1}$ where $X_0 = \{1\}$, $X_{2m+1} = \{n\}$ (with $n = 4m + 2$) and $X_i = \{x_{i,1}, x_{i,2}\}$. There are three categories of arcs in $U$ defined as follows :

- $U_1 = \{(1, x_{1,1}), (1, x_{1,2}), (x_{2m,1}, n), (x_{2m,2}, n)\}$, each arc $(i, j)$ in $U_1$ is valued by $\{0, 0\}$
- $U_2 = \bigcup_{k=1}^{m-1} \{(x_{2k,i}, x_{2k+1,j}) / i = 1, 2 \ ; \ j = 1, 2\}$, each arc $(i, j)$ in $U_2$ is also valued by $\{0, 0\}$
- $U_3 = \{(x_{2k-1,i}, x_{2k,i}) / i = 1, 2 \ ; \ k = 1, \ldots, m\}$, each arc in $U_3$ is valued in a different way :

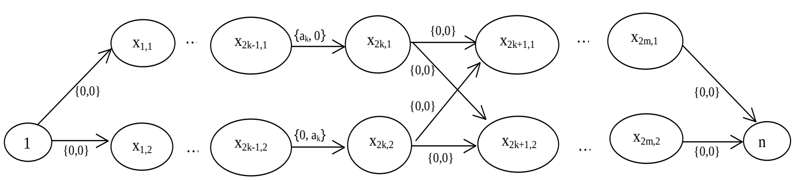



FIG. 5 – Reduction of the 2-partition problem to the ROBDISWOR problem

Considering a path $\mu$ from 1 to $n$ in $G$, let us set :

$$I' = \{k / (x_{2k-1,1}, x_{2k,1}) \in \mu \ \forall k = 1, \ldots, m\}$$
$$I \setminus I' = \{k / (x_{2k-1,2}, x_{2k,2}) \in \mu \ \forall k = 1, \ldots, m\}$$

Thus $c^1(\mu) = \sum_{k \in I'} a_k$ and $c^2(\mu) = \sum_{k \in I \setminus I'} a_k$. Consequently, a 2-partition exists if and only if there exists a path $\mu$ from 1 to $n$ in $G$ such as $v_{\text{PIR}}(\mu) = \max\{c^1(\mu), c^2(\mu)\} = \frac{\sum_{k \in I} a_i}{2}$, this value being the optimal path value $\mu^*_{\text{PIR}}$.

### 5.1.2 Resolution

Yu and Yang in [27] propose an exact algorithm for solving ROBDISWOR when the number of scenarios is bounded. It is based on the dynamic programming principle with a pseudo-polynomial time complexity. In particular for layered graphs, the complexity is $O(|U|(L_{\max})^q)$ where $L_{\max} = \max_{s\in S} L_s$ with $L_s$ denoted the length of the longest path from $1$ to $n$ for the scenario $s$. Yu and Yang also propose in [27] an approximate algorithm, denoted $H$. For layered graphs, the complexity is $O(|U|q)$. This algorithm consists to compute the shortest path for each scenario and, for a fictitious scenario which is an average of the $q$ scenarios, to determine the path which minimizes the worst case criterion. Given $\mu^h_{\text{WOR}}$ the path determined by $H$, the authors show that :

$$\frac{v_{\text{WOR}(\mu^H_{\text{WOR}})}}{v_{\text{WOR}(\mu^*_{\text{WOR}})}} \le \frac{q\frac{\max_{s\in S} c^s(\mu^H_{\text{WOR}})}{\min_{s\in S} c^s(\mu^H_{\text{WOR}})}}{\frac{\max_{s\in S} c^s(\mu^H_{\text{WOR}})}{\min_{s\in S} c^s(\mu^H_{\text{WOR}})} + q - 1}$$

## 5.2 With the maximum regret criterion

Yu and Yang in [27] show that the 2-partition problem is reduced to ROBDISREG, for which the problem is to decide if there is a path $\mu$ from 1 to $n$ such as $v_{\text{REG}}(\mu) = v$. The reduction presented in the previous section remains valid since $v_{\text{REG}}(\mu) = v_{\text{WOR}}(\mu)$ because $c^1(\mu^*_{G^1}) = c^2(\mu^*_{G^2}) = 0$. In addition, exact and approximate algorithms for solving ROBDISWOR can be applied, after some minor modifications for solving ROBDISREG.

## 5.3 With the multicriteria methodology

In the discrete set of scenarios model, the problem of determining all non-dominated paths can be solved by applying algorithms defined in multicriteria analysis as suggested in [8]. Two types of algorithms are proposed in the literature for determining all non-dominated paths in a multicriteria graph : graph algorithms which generalize classical monocriterion graph algorithms to the multicriteria case and, those based on the enumeration of the $k$ shortest paths. The efficiency of this approach rapidly decreases when the number of scenarios increases since the number of non-dominated paths significantly increases with the number of scenarios.

Concerning the problem to determine all Lorenz non-dominated paths, Perny and Spanjaard propose in [21] an $A^*$ algorithm.

# 6 Conclusion

Robust path problem has been extensively studied. For most of the considered models, robust version of the shortest path problem becomes NP-difficult, excepted for the worst case analysis with the interval model which is tractable in polynomial time. However, the worst case analysis supposes that the scenario that will be realized will adversely affect the solution. This hypothesis may not be relevant in some decision context. In such cases, new models of robustness have to be proposed.

Concerning the modeling of uncertain, undetermined and random factors, it can be relevant in some context to consider them on graph structure and not only on arcs' value. In such a model, a scenario may be a particular partial subgraph. Such a "structural" model can be seen as an extension of probabilistic combinatorial optimization (cf [20] for the study of the longest probabilistic path).

# References

[1] H. Aissi, C. Bazgan, and D. Vanderpooten. Complexity of the min-max and min-max regret assignment problems. *Operations Research Letters*, 33(6):634–640, 2005.

[2] I. Averbakh. Minmax regret solutions for minimax optimization problems with uncertainty. *Operations Research Letters*, 27:57–65, 2000.

[3] I. Averbakh. On the complexity of a class of combinatorial optimization problems with uncertainty. *Math. Program., Ser. A*, 90:263–272, 2001.

[4] I. Averbakh and V. Lebedev. Interval data minmax regret network optimization problems. *Discrete Applied Mathematics*, 138:289–301, 2004.

[5] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Math. Program., Ser. B*, 98:49–71, 2003.

[6] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[7] V. G. Deĭneko and G. J. Woeginger. On the robust assignment problem under a fixed number of cost scenarios. *Operations Research Letters*, 34(2):175–179, 2006.

[8] L. C. Dias and J. N. Clímaco. Shortest path problems with partial information: models and algorithms for detecting dominance. *European Journal of Operational Research*, 121:16–31, 2000.

[9] M.RM Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co., San Francisco, 1979.

[10] P. Hansen. Bicriterion path problems. In *Multicriteria Decision making*. In G. Fandel and T. Gal (eds), 1980.

[11] R. Hites, Y. De Smet, N. Risse, M. Salazar-Neumann, and P. Vincke. About the applicability of mcda to some robustness problems. *European Journal of Operational Research*, 174:322–332, 2006.

[12] R. Kalai, M.A. Aloulou, P. Vallin, and D. Vanderpooten. Robust 1-median location problem on a tree. In *Proceedings of the ORP3 meeting*, Valencia, Espagne, September 2005.

[13] O. E. Karaşan, M. Ç. Pinar, and H. Yaman. The robust shortest path problem with interval data. Technical report, Bilkent University, 2001.

[14] A. Kasperski and P. Zieliński. The robust shortest path problem in series-parallel multidigraphs with interval data. *Operations Research Letters*, 34(1):69–76, 2006.

[15] P. Kouvelis and G. Yu. *Robust dicrete optimization and its applications*. Kluwer Academic Publishers, Boston, 1997.

[16] R Montemanni. A benders decomposition approach for the robust spanning tree problem with interval data. *European Journal of Operational Research*, 174(3):1479–1490, 2006.

[17] R. Montemanni and L.M. Gambardella. An exact algorithm for the robust shortest path problem with interval data. *Computers and Operations Research*, 31:1667–1680, 2004.

[18] R. Montemanni and L.M. Gambardella. The robust path problem with interval data via benders decomposition. *4OR*, 3(4):315–328, 2005.

[19] R. Montemanni, L.M. Gambardella, and A.V. Donati. A branch and bound algorithm for the robust shortest path problem with interval data. *Operations Research Letters*, 32:225–232, 2004.

[20] C. Murat and V. Paschos. The probabilistic longest path problem. *Networks*, 33:207–219, 1999.

[21] P. Perny and O. Spanjaard. An axiomatic approach to robustness in search problems with multiple scenarios. In *Proceedings of the 19th conference on Uncertainty in Artificial Intelligence*, pages 469–476, Acapulco, Mexico, 2003.

[22] B. Roy. *Méthodologie multicritère d'aide à la décision*. Economica, Paris, 1985.

[23] B. Roy. *Flexibilité et Robustesse en Ordonnancement*, chapter A propos de robustesse en recherche Opérationnelle et Aide à la Décision. Hermès, Paris, 2004.

[24] O. Spanjaard. *Exploitation de préférences non-classiques dans les problèmes combinatoires : modèles et algorithmes pour les graphes*. Thèse de doctorat, Université Paris Dauphine - LAMSADE, 2003.

[25] P. Vincke. Robust solutions and methods in decision-aid. *Journal of multi-criteria decision analysis*, 8:181–187, 1999.

[26] H. Yaman, O. E. Karaşan, and M. Ç. Pinar. The robust spanning tree problem with interval data. *Operations Research Letters*, 29:31–40, 2001.

[27] G. Yu and J. Yang. On the robust shortest path problem. *Computers and Operations Research*, 25(6):457–468, 1998.

[28] P. Zieliński. The computational complexity of the relative robust shortest path problem with interval data. *European Journal of Operational Research*, 158:570–576, 2004.

# Robust multiple criteria ranking using a set of additive value functions

Salvatore Greco*, Vincent Mousseau†, Roman Słowiński*

**Résumé**

Nous présentons une nouvelle méthode, appelée UTA$^{\text{GMS}}$, pour le problème de rangement multicritère d'un ensemble d'actions $A$ sur la base d'un ensemble de fonctions de valeurs additives calculé par régression ordinale. L'information préférentielle obtenue du décideur est un ensemble de comparaisons par paires sur un sous-ensemble $A^R \subseteq A$ d'actions, appelées actions de référence. Le modèle de préférence construit par régression ordinale est l'ensemble de toutes les fonctions de valeur additives compatibles avec l'information préférentielle. En utilisant ce modèle, il est possible de définir deux relations sur $A$ : une relation de surclassement nécessaire qui est vraie pour deux actions $a, b \in A$ ssi pour toute fonction de valeur compatible $U$ on a $U(a) \geq U(b)$, et la relation de surclassement possible qui est vraie pour deux actions $a, b \in A$ ssi il existe au moins une fonction de valeur compatible $U$ telle que $U(a) \geq U(b)$. Ces deux relations définissent un classement possible et nécessaire des actions de $A$, et sont, respectivement, un préordre partiel, et une relation fortement complète et négativement transitive. La méthode UTA$^{\text{GMS}}$ est conçue pour être utilisée de façon interactive, avec un ensemble $A^R$ qui s'accroît de sorte que le décideur ajoute progressivement des comparaisons par paire. Lorsqu'aucune comparaison pas paire n'est fournie, le surclassement nécessaire correspond à la relation de dominance et le surclassement possible est une relation complète. Chaque nouvelle comparaison d'actions de référence ne correspondant pas à une situation de dominance enrichit la relation de surclassement nécessaire et appauvrit la relation de surclassement possible, de sorte qu'elles convergent à mesure que l'information préférentielle s'enrichit. En distinguant les conséquences nécessaires et possibles sur $A$

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

‡Faculty of Economics, University of Catania, Corso Italia, 55, 95129 Catania, Italy, salgreco@mbox.unict.it

§Institute of Computing Science, Poznań University of Technology, 60-965 Poznań, and Institute for Systems Research, Polish Academy of Sciences, 01-447 Warsaw, Poland, roman.slowinski@cs.put.poznan.pl

d'informations de préférence, UTA$^{\text{GMS}}$ répond à des questions relevant de l'analyse de robustesse. De plus, la méthode aide le décideur aussi lorsque ses préférences ne peuvent pas être représentées en termes de fonctions de valeurs additives. La méthode est illustrée sur un exemple résolu avec le logiciel VisualUTA v1.0. Des extensions sont aussi proposées.

**Mots-clefs :** Analyse de robustesse, Rangement multicritère, Régression ordinale, Fonction de valeur additive

**Abstract**

We present a new method, called UTA$^{\text{GMS}}$, for multiple criteria ranking of alternatives from set $A$ using a set of additive value functions which result from an ordinal regression. The preference information provided by the decision maker is a set of pairwise comparisons on a subset of alternatives $A^R \subseteq A$, called reference alternatives. The preference model built via ordinal regression is a set of all additive value functions compatible with the preference information. Using this model, one can define two relations in the set $A$: the necessary weak preference relation which holds for any two alternatives $a, b$ from set $A$ if and only if for all compatible value functions $a$ is preferred to $b$, and the possible weak preference relation which holds for this pair if and only if for at least one compatible value function $a$ is preferred to $b$. These relations establish a necessary and a possible ranking of alternatives from $A$, being, respectively, a partial preorder and a strongly complete and negatively transitive relation. The UTA$^{\text{GMS}}$ method is intended to be used interactively, with an increasing subset $A^R$ and a progressive statement of pairwise comparisons. When no preference information is provided, the necessary weak preference relation is a weak dominance relation, and the possible weak preference relation is a complete relation. Every new pairwise comparison of reference alternatives, for which the dominance relation does not hold, is enriching the necessary relation and it is impoverishing the possible relation, so that they converge with the growth of the preference information. Distinguishing necessary and possible consequences of preference information on the all set of actions, UTA$^{\text{GMS}}$ answers questions of robustness analysis. Moreover, the method can support the decision maker also when his/her preference statements cannot be represented in terms of an additive value function. The method is illustrated by an example solved using the UTA$^{\text{GMS}}$ software. Some extensions of the method are also presented.

**Key words :** Robustness analysis, Multiple criteria ranking, Ordinal regression approach, Additive value function

# 1 Introduction

We are considering a decision situation in which a finite set of alternatives (actions) $A$ is evaluated on a family of $n$ criteria $g_1, g_2, \ldots, g_i, \ldots, g_n$, with $g_i : A \to \mathbb{R}$ for all

$i \in G = \{1, 2, \ldots, n\}$. We assume, without loss of generality, that the greater $g_i(a)$, the better alternative $a$ on criterion $g_i$, for all $i \in G$. A decision maker (DM) is willing to rank the alternatives in $A$ from the best to the worst, according to his/her preferences. The ranking can be complete or partial, depending on the preference information supplied by the DM and on the way of using this information. The family of criteria $G$ is supposed to satisfy the following consistency conditions (see [24]):

- exhaustivity - any two alternatives having the same evaluations on all criteria from $G$ should be considered indifferent,
- monotonicity - when comparing two alternatives, an improvement of one of them on at least one criterion from $G$ should not deteriorate its comparison to the other alternative,
- non-redundancy - deletion of any criterion from $G$ will contradict one of the two above conditions.

Such a decision problem is called multiple criteria ranking problem. It is known that the only information coming out from the formulation of this problem is the weak dominance relation. Let us recall that the weak dominance relation is a partial preorder: alternative $a \in A$ is preferred to alternative $b \in A$ (denotation $a \succ b$) if and only if $g_i(a) \geq g_i(b)$ for all $i \in G$, with at least one strict inequality; moreover, $a$ is indifferent to $b$ (denotation $a \sim b$) if and only if $g_i(a) = g_i(b)$ for all $i \in G$; hence, for any two alternatives $a, b \in A$, one of the four situations may arise in the weak dominance relation: $a \succ b$, $a \prec b$, $a \sim b$ and $a?b$, where the last one means that $a$ and $b$ are incomparable. Usually, the weak dominance relation is very poor, i.e. the most frequent situation is $a?b$.

In order to enrich the weak dominance relation, multiple criteria decision aiding (MCDA) helps in construction of an aggregation model on the base of preference information provided by the DM. Such an aggregation model is called preference model - it induces a preference structure in set $A$ whose proper exploitation permits to work out a ranking proposed to the DM.

The preference information may be either direct or indirect, depending if it specifies directly values of some parameters used in the preference model (e.g. trade-off weights, aspiration levels, discrimination thresholds, etc.), or if it specifies some examples of holistic judgments from which compatible values of the preference model parameters are induced. Direct preference information is used in the traditional aggregation paradigm, according to which the aggregation model is first constructed and then applied on set $A$ to rank the alternatives.

Indirect preference information is used in the disaggregation (or regression) paradigm, according to which the holistic preferences on a subset of alternatives $A^R \subseteq A$ are known first, and then a consistent aggregation model is inferred from this information to be applied on set $A$ in order to rank the alternatives.

Presently, MCDA methods based on indirect preference information and the disaggregation paradigm are of increasing interest for they require relatively less cognitive effort from the DM. Indeed, the disaggregation paradigm is consistent with the "posterior rationality" postulated by March [18] and with the inductive learning used in artificial intelligence approaches (see [19]). Typical applications of this paradigm in MCDA are presented in [29], [22], [10], [13], [1], [21], [7], [8], [9].

In this paper, we are considering the aggregation model in form of an additive value function:

$$U(a) = \sum_{i=1}^{n} u_i(a) \tag{1}$$

where $u_i(a) \geq 0,\ i = 1, \ldots, n$, are nondecreasing marginal value functions. While the additive value function involves compensation between criteria and requires rather strong assumption that criteria are independent in the sense of preferences [11], it is often used for its intuitive interpretation and relatively easy computation. The weighted-sum aggregation model, which is a particular case of the additive value function, is used even more frequently in spite of its simplistic form (see e.g. [1], [25]).

We are using the additive aggregation model in the settings of the disaggregation paradigm, as it has been proposed in the UTA method (see [10]). In fact, our method generalizes the UTA method in two aspects:

- it takes into account all additive value functions (1) compatible with indirect preference information, while UTA is using only one such function,
- the marginal value functions of (1) are general non-decreasing functions, and not piecewise linear, as in UTA,
- the DM's ranking of reference alternatives does not need to be complete.

The preference information used by our method is provided in the form of a set of pairwise comparisons of some alternatives from a subset $A^R \subseteq A$, called reference alternatives. The method is producing two rankings in the set of alternatives $A$, such that for any pair of alternatives $a, b \in A$:

- in the *necessary* ranking, $a$ is ranked at least as good as $b$ if and only if, $U(a) \geq U(b)$ for all value functions compatible with the preference information,
- in the *possible* ranking, $a$ is ranked at least as good as $b$ if and only if, $U(a) \geq U(b)$ for at least one value function compatible with the preference information.

The necessary ranking can be considered as robust with respect to the indirect preference information. Such robustness of the necessary ranking refers to the fact that any pair of alternatives compares in the same way whatever the additive value function compatible with the indirect preference information. Indeed, when no indirect preference information is given, the necessary ranking boils down to the weak dominance relation, and the possible ranking is a complete relation. Every new pairwise comparison of reference alternatives, for which the dominance relation does not hold, is enriching the necessary ranking and it is impoverishing the possible ranking, so that they converge with the growth of the preference information.

Another appeal of such an approach stems from the fact that it gives space for interactivity with the DM. Presentation of the necessary ranking, resulting from an indirect preference information provided by the DM, is a good support for generating reactions from the DM. Namely, (s)he could wish to enrich the ranking or to contradict a part of it. This reaction can be integrated in the indirect preference information in the next iteration.

The organization of the paper is the following. In the next section, we will outline the principle of the ordinal regression via linear programming, as proposed in the original UTA method (see [10]). In section 3, we give a brief overview of existing approaches to multiple criteria ranking using a set of additive value functions, and we provide motivations for our approach. The new UTA$^{\text{GMS}}$ method is presented in section 4. Some extensions are considered in section 5. Section 6 provides an illustrative example showing how the method can be applied in practice. The last section includes conclusions.

## 2 Ordinal regression via linear programming - principle of the UTA method

Let $X_i$ denote the evaluation scale of criterion $g_i$, $i \in G$. Consequently, $X = \prod_{i=1}^{n} X_i$ is the evaluation space, and $x, y \in X$ denote profiles of alternatives in this space. We consider a weak preference (outranking) relation $\succsim$ on $X$ that states for each pair of vectors $x, y \in X$: $x \succsim y \;\Leftrightarrow\;$ "$x$ is at least as good as $y$". This weak preference relation can be decomposed into its asymmetric and symmetric parts, as follows:

- $x \succ y \;\Leftrightarrow\; [x \succsim y$ and not$(y \succsim x)] \;\Leftrightarrow\;$ "$x$ is preferred to $y$",
- $x \sim y \;\Leftrightarrow\; [x \succsim y$ and $y \succsim x] \;\Leftrightarrow\;$ "$x$ is indifferent to $y$"

>From a pragmatic viewpoint, it is reasonable to assume that $X_i = [\alpha_i, \beta_i]$, i.e. the evaluation scale on each criterion $g_i$ is bounded, such that $\alpha_i < \beta_i$ are the worst and the best (finite) evaluations, respectively. Thus, $g_i:\ A \mapsto X_i,\ i \in G$, therefore, each alternative $a \in A$ is associated with an evaluation vector denoted by $\boldsymbol{g}(a) \in X$.

The additive value function is defined on $X$ such that for each $a \in A$

$$U(\boldsymbol{g}(a)) = \sum_{i=1}^{n} u_i(g_i(a)), \tag{2}$$

where $u_i$ are non-decreasing marginal value functions, $u_i : X_i \mapsto \mathbb{R},\ i = 1, \ldots, n$. For simplicity, we will write (2) as (1), i.e. $U(a) = \sum_{i=1}^{n} u_i(a)$.

In the following, we recall the principle of the UTA method as presented recently in [28]. The indirect preference information is given in the form of a complete preorder on a subset of reference alternatives $A^R \subseteq A$, called reference preorder. The reference alternatives are usually those alternatives in set $A$ for which the DM is ready to express holistic preferences. Let the set of reference alternatives $A^R = \{a_1, a_2, ..., a_m\}$ be rearranged such that $a_k \succsim a_{k+1},\ k = 1, ..., m-1$, where $m = |A^R|$. The disaggregation paradigm consists here in inferring an additive value function (1) ranking the reference alternatives in exactly the same way as it was done by the DM. Such a value function $U$ is called *compatible*.

The ordinal regression consists in the inference of a compatible value function restoring the reference preorder. The transition from a reference preorder to a value function is done according to the following equivalence :

$$\begin{array}{l} U(a_k) > U(a_{k+1}) \;\Leftrightarrow\; a_k \succ a_{k+1} \\ U(a_k) = U(a_{k+1}) \;\Leftrightarrow\; a_k \sim a_{k+1} \end{array} \tag{3}$$

for $k = 1, ..., m-1$.

In the UTA method, the marginal value functions $u_i$ are assumed to be piecewise linear, so that the intervals $[\alpha_i, \beta_i]$ are divided into $\gamma_i \geq 1$ equal sub-intervals: $[x_i^0, x_i^1]$, $[x_i^1, x_i^2]$, ..., $[x_i^{\gamma_i - 1}, x_i^{\gamma_i}]$, where $x_i^j = \alpha_i + \frac{j(\beta_i - \alpha_i)}{\gamma_i}$, $j = 0, \ldots, \gamma_i$, $i = 1, \ldots, n$. The marginal value (see Figure 1) of an alternative $a \in A$ is approximated by linear interpolation

$$u_i(a) = u_i(x_i^j) + \frac{g_i(a) - x_i^j}{x_i^{j+1} - x_i^j} \left( u_i(x_i^{j+1}) - u_i(x_i^j) \right), \text{ for } g_i(a) \in [x_i^j, x_i^{j+1}] \tag{4}$$

According to (4), the piecewise linear additive model is completely defined by the marginal values at the characteristic points, i.e. $u_i(x_i^0) = u_i(\alpha_i),\ u_i(x_i^1),\ u_i(x_i^2), \ldots,$ $u_i(x_i^{\gamma_i}) = u_i(\beta_i)$.

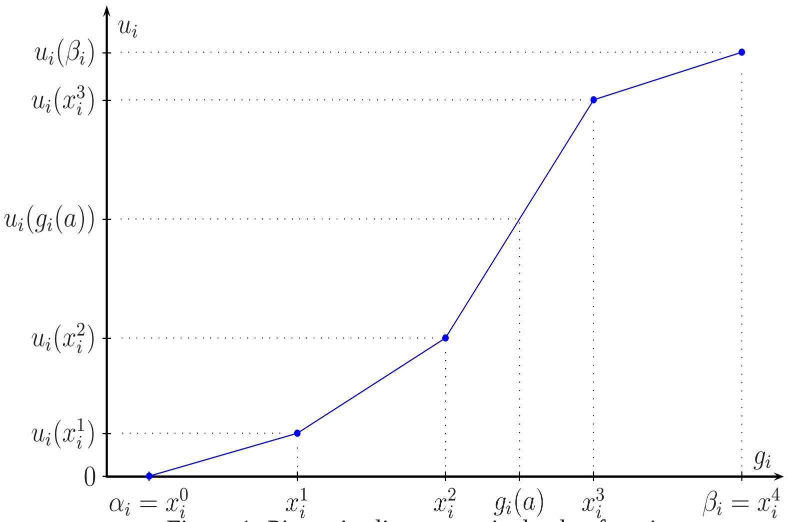


Figure 1: Piecewise linear marginal value function

It is also usual to suppose a kind of normalization such as $u_i(\alpha_i) = 0,\ \forall i \in G$, and $\sum_{i=1}^n u_i(\beta_i) = 1$. This will bound the value function $U(a)$ in the interval [0,1].

Therefore, a value function $U(a) = \sum_{i=1}^n u_i(a)$ is compatible if it satisfies the following set of constraints

$$\begin{array}{l} \left. \begin{array}{l} U(a_k) > U(a_{k+1}) \Leftrightarrow\ a_k \succ a_{k+1} \\ U(a_k) = U(a_{k+1}) \Leftrightarrow\ a_k \sim a_{k+1} \end{array} \right\} \\ k = 1, \ldots, m-1 \\ u_i(x_i^{j+1}) - u_i(x_i^j) \geq 0,\ i = 1, ..., n,\ j = 1, ..., \gamma_i - 1 \\ u_i(\alpha_i) = 0,\ i = 1, ..., n \\ \sum_{i=1}^n u_i(\beta_i) = 1 \end{array} \tag{5}$$

To verify if a compatible value function $U(a) = \sum_{i=1}^{n} u_i(a)$ restoring the reference preorder $\succsim$ on $A^R$ exists, one can solve the following linear programming problem, where $u_i(x_i^j), i = 1, ..., n,\ j = 1, ..., \gamma_i$ are unknown, and $\sigma^+(a), \sigma^-(a), a \in A^R$ are auxiliary variables:

$$
\begin{array}{l}
Min \rightarrow \quad F = \sum_{k=1}^{m} \left(\sigma^+(a_k) + \sigma^-(a_k)\right) \\
\qquad \text{s.t.} \\
\qquad\quad \left.\begin{array}{l}
U(a_k) + \sigma^+(a_k) - \sigma^-(a_k) \geq \\
\qquad U(a_{k+1}) + \sigma^+(a_{k+1}) - \sigma^-(a_{k+1}) + \varepsilon \ \Leftrightarrow \ a_k \succ a_{k+1} \\
U(a_k) + \sigma^+(a_k) - \sigma^-(a_k) = \\
\qquad U(a_{k+1}) + \sigma^+(a_{k+1}) - \sigma^-(a_{k+1}) \ \Leftrightarrow \ a_k \sim a_{k+1}
\end{array}\right\} k = 1, \ldots, m-1 \\
\qquad\quad u_i(x_i^{j+1}) - u_i(x_i^j) \geq 0,\ i = 1, ..., n,\ j = 1, ..., \gamma_i - 1 \\
\qquad\quad u_i(\alpha_i) = 0,\ i = 1, ..., n \\
\qquad\quad \sum_{i=1}^{n} u_i(\beta_i) = 1 \\
\qquad\quad \sigma^+(a_k),\ \sigma^-(a_k) \geq 0,\ k = 1, ..., m
\end{array}
\tag{6}
$$

where $\varepsilon$ is an arbitrarily small positive value so that $U(a_k)+\sigma^+(a_k)-\sigma^-(a_k) > U(a_{k+1})+\sigma^+(a_{k+1}) - \sigma^-(a_{k+1})$ in case of $a_k \succ a_{k+1}$.

If the optimal value of the objective function of the program (6) is equal to zero ($F^* = 0$), then there exists at least one value function $U(a) = \sum_{i=1}^{n} u_i(a)$ satisfying (5), i.e. compatible with the reference preorder on $A^R$. In other words, this means that the corresponding polyhedron (5) of feasible solutions for $u_i(x_i^j),\ i = 1, ..., n,\ j = 1, ..., \gamma_i$ is not empty.

Let us remark that the transition from the preorder $\succsim$ to the marginal value function exploits the ordinal character of the criterion scale $X_i$. Note, however, that the scale of the marginal value function is a conjoint interval scale. More precisely, for the considered additive value function, the admissible transformations on the marginal value functions $u_i(x_i)$ have the form $u_i^*(x_i) = k \times u_i(x_i) + h_i$, $h_i \in \mathbb{R},\ i = 1, ..., n$, $k > 0$, such that for all $[x_1, ..., x_n], [y_1, ..., y_n] \in \prod_{i=1}^{n} X_i$

$$\sum_{i=1}^{n} u_i(x_i) \geq \sum_{i=1}^{n} u_i(y_i) \Leftrightarrow \sum_{i=1}^{n} u_i^*(x_i) \geq \sum_{i=1}^{n} u_i^*(y_i).$$

An alternative way of representing the same preference model is:

$$U(a) = \sum_{i=1}^{n} w_i \hat{u}_i(a), \quad \text{where } \hat{u}(\alpha_i) = 0,\ \hat{u}(\beta_i) = 1,\ w_i \geq 0\ \ \forall i \in G, \text{ and } \sum_{i=1}^{n} w_i = 1. \tag{7}$$

Note that the correspondence between (7) and (1) is such that $w_i = u_i(\beta_i),\ \forall i \in G$. Due to the cardinal character of the marginal value function scale, the parameters $w_i$ can be interpreted as tradeoff weights among marginal value functions $\hat{u}_i(a)$. We will use, however, the preference model (1) with normalization constraints bounding $U(a)$ to the interval $[0, 1]$.

When the optimal value of the objective function of the program (6) is greater than zero ($F^* > 0$), then there is no value function $U(a) = \sum_{i=1}^{n} u_i(a)$ compatible with the reference preorder on $A^R$. In such a case, three possible moves can be considered:

- increasing the number of linear pieces $\gamma_i$ for one or several marginal value function $u_i$ could make it possible to find an additive value function compatible with the reference preorder on $A^R$,
- revising the reference preorder on $A^R$ could lead to find an additive value function compatible with the new preorder,
- searching over the relaxed domain $F \leq F^* + \eta$ could lead to an additive value function giving a preorder on $A^R$ sufficiently close to the reference preorder (in the sense of Kendall's $\tau$).

# 3 Existing approaches and motivations for a new method

Our work aims at generalizing the UTA method in order to consider the set of all value functions compatible with the indirect preference information rather than choosing a single value function within the set of compatible ones. The literature concerning MCDA methods involving a set of additive value functions can be viewed from three points of view:

- The methods are designed for different problem statements (problematics, see [23]):
  - choice of the best alternative (*e.g.* [2], [14], [16], [5], [26]),
  - sorting alternatives into predefined categories (*e.g.* [15], [4]),
  - ranking of alternatives from the best to the worst (*e.g.* [10], [12])
- The methods also differ with respect to the kind of the set of value functions and the characteristics of these functions: linear (*e.g.* [14], [12]) or piecewise linear (*e.g.* [3], [10], [4]) or monotone (*e.g.* [1]) value functions.
- The sets of value functions can be:

- explicitly listed (*e.g.* [27]),
- defined from stated constraints on the functions (*e.g.* [2], [17]),
- induced from holistic preference statements concerning alternatives (*e.g.* [15], [31], [1]).

A review of the literature and, particularly, of the methods based on the ordinal regression approach, shows that these methods fail to consider some important issues :

- If the polyhedron of value functions compatible with the stated preference information is not empty, then the choice of a single or few representative value functions is either arbitrary or left to the DM. In the latter case, the DM is supposed to know how to interpret the form of the marginal value functions in order to choose among them, which is not easy for most DMs. Therefore, it seems reasonable to accept existence of all value functions compatible with the preference information provided by the DM and to assess a preference relation in the set of alternatives $A$ with respect to all these functions.

- In most methods, the class of value functions is limited to linear or piecewise linear marginal value functions. To specify the number of characteristic points (breakpoints) is arbitrary and restrictive. It is desirable to consider just monotone marginal value functions which do not involve any parametrization.

- Most methods require that the DM provides constraints on the range of weights of linear marginal value functions, or on the range of variation of piecewise linear marginal value functions. The DM may have, however, difficulties to analyze the link between a specific value function and the resulting ranking. This is why we believe that the DM should be allowed to express preference information in terms of pairwise comparisons of alternatives rather than fixing the above constraints. Providing preference information in this way is consistent with intuitive reasoning of DMs.

- The methods based on ordinal regression are usually considering the preference information provided by the DM as a whole. As a consequence, it is difficult for the DM to associate a piece of his/her preference information with the result and, therefore, to control the impact of each piece of information (s)he provides on the result. As such a control is desirable for a truly interactive process, ordinal regression methods should allow the DM to provide incrementally the preference information by possibly small pieces.

In this paper, we intend to present a new ordinal regression method that accounts for all shortcomings listed above.

# 4 The new UTA$^{\text{GMS}}$ method

## 4.1 Presentation of the method

The new UTA$^{\text{GMS}}$ method is an ordinal regression method using a set of additive value functions $U(a) = \sum_{i=1}^{n} u_i(g_i(a))$ as a preference model. One of its characteristic features is that it takes into account the set of all value functions compatible with the preference information provided by the DM. Moreover, it considers general non-decreasing marginal value functions instead of piecewise linear only.

We suppose the DM provides preference information in form of pairwise comparisons of reference alternatives from $A^R \subseteq A$. This preference information is a partial preorder on $A^R$, denoted by $\succsim$. A value function is called *compatible* if it is able to restore the partial preorder $\succsim$. Each compatible value function induces, moreover, a ranking on the whole set $A$.

In particular, for any two alternatives $x, y \in A$, a compatible value function ranks $x$ and $y$ in one of the following ways: $x \succ y$, $y \succ x$, $x \sim y$. With respect to $x, y \in A$, it is thus reasonable to ask the following two questions:

- are $x$ and $y$ ranked in the same way by <u>all</u> compatible value functions?
- is there <u>at least one</u> compatible value function ranking $x$ at least as good as $y$ (or $y$ at least as good as $x$)?

Having answers to these questions for all pairs of alternatives $(x, y) \in A \times A$, one gets a *necessary* weak preference relation $\succsim^N$, in case $U(x) \geq U(y)$ for all compatible value functions, and a *possible* weak preference relation $\succsim^P$ in $A$, in case $U(x) \geq U(y)$ for at least one compatible value function.

Let us remark that preference relations $\succsim^N$ and $\succsim^P$ are meaningful only if there exists at least one compatible value function. Observe also that in this case, for any $x, y \in A^R$,

$$x \succsim y \Rightarrow x \succsim^N y$$

and

$$x \succ y \Rightarrow \text{not } (y \succsim^P x).$$

In fact, if $x \succsim y$, then for any compatible value function $U(x) \geq U(y)$ and, therefore, $x \succsim^N y$. Moreover, if $x \succ y$, then for any compatible value function $U(x) > U(y)$ and, consequently, there is no compatible value function such that $U(y) \geq U(x)$, which means that not $y \succsim^P x$.

Formally, a general additive compatible value function is an additive value function $U(a) = \sum_{i=1}^{n} u_i(a)$ satisfying the following set of constraints:

$$\left.\begin{array}{l} \left.\begin{array}{l} U(a) > U(b) \;\Leftrightarrow\; a \succ b \\ U(a) = U(b) \;\Leftrightarrow\; a \sim b \end{array}\right\} \forall a,b \in A^R \\ u_i(g_i(a_{\tau_i(j)})) - u_i(g_i(a_{\tau_i(j-1)})) \geq 0,\; i=1,...,n,\; j=2,...,m \\ u_i(g_i(a_{\tau_i(1)})) \geq 0, u_i(g_i(a_{\tau_i(m)})) \leq u_i(\beta_i),\; i=1,...,n, \\ u_i(\alpha_i) = 0,\; i=1,...,n \\ \sum_{i=1}^{n} u_i(\beta_i) = 1, \end{array}\right\} (E^{A^R})$$

where $\tau_i$ is the permutation on the set of indices of alternatives from $A^R$ that reorders them according to the increasing evaluation on criterion $g_i$, i.e.

$$g_i(a_{\tau_i(1)}) \leq g_i(a_{\tau_i(2)}) \leq \ldots \leq g_i(a_{\tau_i(m-1)}) \leq g_i(a_{\tau_i(m)})$$

Remark that, due to this formulation of the ordinal regression problem, no linear interpolation is required to express the marginal value of any reference alternative. Thus, one cannot expect that increasing the number of characteristic points will bring some "new" compatible additive value functions. In consequence, UTA$^{\text{GMS}}$ considers all compatible additive value functions while classical UTA ordinal regression (6) deals with a subset of the whole set of compatible additive value functions, more precisely the subset of piecewise linear additive value functions relative to the considered characteristic points.

## 4.2 Properties of the relations $\succsim^N$ and $\succsim^P$

Binary relations $\succsim^N$ and $\succsim^P$ satisfy the following interesting properties.

**Proposition 4.1.** $\succsim^P \supseteq \succsim^N$

*Proof.* If $U(x) \geq U(y)$ for all compatible value functions $U$, i.e. $x \succsim^N y$, then there is at least one compatible value function $U'$ such that $U'(x) \geq U'(y)$, i.e. $x \succsim^P y$. □

**Proposition 4.2.** *For all $x, y \in A$, $x \succsim^N y$ or $y \succsim^P x$.*

*Proof.* Let us denote by $\mathcal{U}$ the set of value functions compatible with $\succsim$. For all $x, y \in A$,

$$U(x) \geq U(y)\ \forall U \in \mathcal{U} \text{ or } \exists U \in \mathcal{U} \text{ such that } U(y) > U(x) \Rightarrow x \succsim^N y \text{ or } y \succsim^P x.$$

□

Let us observe that from Proposition 4.2, we can get the following interesting corollary.

**Corollary 4.1.** *For all $x, y \in A$,*

*1) not $(x \succsim^N y) \Rightarrow y \succsim^P x$,*

*2) not $(x \succsim^P y) \Rightarrow y \succsim^N x$.*

*Proof.* 1) Since $x \succsim^N y$ or $y \succsim^P x$, if not $(x \succsim^N y)$, then $y \succsim^P x$.

2) Since $x \succsim^P y$ or $y \succsim^N x$, if not $(x \succsim^P y)$, then $y \succsim^N x$.

□

**Proposition 4.3.** *$\succsim^N$ is a partial preorder (i.e. reflexive and transitive)*

*Proof.* For all $x \in A$, $U(x) = U(x)$. This is true also for all $U$ being compatible value functions, such that $x \succsim^N x$. Let us suppose that for $x, y, z \in A$, we have $x \succsim^N y$ and $y \succsim^N z$. This means that for all compatible value functions $U$ we have $U(x) \geq U(y)$ and $U(y) \geq U(z)$, which implies that for all compatible value functions $U$ we have $U(x) \geq U(z)$, i.e. $x \succsim^N z$. □

**Proposition 4.4.** *$\succsim^P$ is strongly complete, i.e. $\forall x, y \in A,\ x \succsim^P y$ or $y \succsim^P x$, and negatively transitive, i.e. $\forall x, y, z \in A,$ not$(x \succsim^P y)$ and not$(y \succsim^P z) \Rightarrow$ not$(x \succsim^P z)$*

*Proof.* Consider any compatible value function $U$. For each pair $x, y \in A$, it holds $U(x) \geq U(y)$ or $U(y) \geq U(x)$, i.e. $x \succsim^P y$ or $y \succsim^P x$; therefore $\succsim^P$ is strongly complete.
$not(x \succsim^P y)$ means that there does not exist any compatible value function $U$ such that $U(x) \geq U(y)$. $not(y \succsim^P z)$ means that there does not exist any compatible value function $U$ such that $U(y) \geq U(z)$. Therefore, there does not exist any compatible value function $U$ such that $U(x) \geq U(z)$, which means that $not(x \succsim^P z)$. □

Observe that while $\succsim^P$ is negatively transitive, it is not necessarily transitive, i.e. it is possible that for $x, y, z \in A$, $x \succsim^P y$, $y \succsim^P z$ but not $x \succsim^P z$. This can happen because there could exist one compatible value function $U$ such that $U(x) \geq U(y)$ and one compatible value function $U'$ such that $U'(y) \geq U'(z)$, however, there could be no compatible value function $U''$ such that $U''(x) \geq U''(z)$.

Notice that it is impossible to infer $\succsim^N$ from $\succsim^P$ or vice versa, since $\succsim^N$ and $\succsim^P$ are not dual, i.e. $x \succsim^N y \Leftrightarrow$ not$(y \succsim^P x)$ does not hold as one could expect. In fact, in case for $x, y \in A$, $U(x) = U(y)$ for all compatible value functions $U$, we have $x \succsim^N y$ and $y \succsim^P x$.

>From the two weak preference relations $\succsim^N$ and $\succsim^P$, one can get preference, indifference and incomparability, in a usual way, i.e.

1) from the necessary weak preference relation $\succsim^N$ one obtains:

- preference: $x \succ^N y \;\Leftrightarrow\; x \succsim^N y$ and not$(y \succsim^N x)$
- indifference: $x \sim^N y \;\Leftrightarrow\; x \succsim^N y$ and $y \succsim^N x$
- incomparability: $x ?^N y \;\Leftrightarrow\;$ not$(x \succsim^N y)$ and not$(y \succsim^N x)$

2) from the possible weak preference relation $\succsim^P$ one obtains:

- preference: $x \succ^P y \;\Leftrightarrow\; x \succsim^P y$ and not$(y \succsim^P x)$
- indifference: $x \sim^P y \;\Leftrightarrow\; x \succsim^P y$ and $y \succsim^P x$

Observe that in case of $\succsim^P$, incomparability is not considered because, for proposition 4.3, $\succsim^P$ is strongly complete.

The preference relations obtained from $\succsim^N$ constitute the necessary ranking, and the preference relations obtained from $\succsim^P$ constitute the possible ranking; they are presented to the DM as end results of the UTA$^{\text{GMS}}$ method at the current stage of interaction.

## 4.3 Computation of the relations $\succsim^N$ and $\succsim^P$

In order to compute binary realtions $\succsim^P$ and $\succsim^N$ we can proceed as follows. For all alternatives $x, y \in A$, let $\pi_i$ be a permutation of the indices of alternatives from set $A^R \cup \{x, y\}$ that reorders them according to increasing evaluation on criterion $g_i$, i.e.

$$g_i(a_{\pi_i(1)}) \leq g_i(a_{\pi_i(2)}) \leq ... \leq g_i(a_{\pi_i(\omega-1)}) \leq g_i(a_{\pi_i(\omega)})$$

where

- if $A^R \cap \{x,y\} = \emptyset$, then $\omega = m+2$
- if $A^R \cap \{x,y\} = \{x\}$ or $A^R \cap \{x,y\} = \{y\}$, then $\omega = m+1$
- if $A^R \cap \{x,y\} = \{x,y\}$, then $\omega = m$.

Then, we can fix the characteristic points of $u_i(g_i)$, $i = 1,...,n$, in

$$g_i^0 = \alpha_i, \quad g_i^j = g_i(a_{\pi_i(j)}) \text{ for } j = 1,...,\omega, \quad g_i^{\omega+1} = \beta_i$$

Let us consider the following set $E(x,y)$ of ordinal regression constraints, with $i = 1,...,n,\ j = 1,...,\omega+1$ as variables:

$$\left.\begin{array}{l} \left.\begin{array}{l} U(a) \geq U(b) + \varepsilon \ \Leftrightarrow \ a \succ b \\ U(a) = U(b) \ \Leftrightarrow \ a \sim b \end{array}\right\} \forall a,b \in A^R \\ u_i(g_i^j) - u_i(g_i^{j-1}) \geq 0,\ i = 1,...,n,\ j = 1,...,\omega+1 \\ u_i(g_i^0) = 0,\ i = 1,...,n \\ \sum_{i=1}^n u_i(g_i^{\omega+1}) = 1, \end{array}\right\} (E(x,y))$$

where $\varepsilon$ is an arbitrarily small positive value as in (6).

The above set of constraints depends on the pair of alternatives $x,y \in A$ because their evaluations $g_i(x)$ and $g_i(y)$ give coordinates for two of $(\omega+1)$ characteristic points of marginal value function $u_i(g_i)$, for each $i = 1,\ldots,n$. Note that for all $x,y \in A$, $E(x,y) = E(y,x)$.

Let us suppose that the polyhedron defined by the set of constraints $E(x,y)$ is not empty. In this case we have that:

$$x \succsim^N y \ \Leftrightarrow \ d(x,y) \geq 0$$

where:

$$d(x,y) = Min\{U(x) - U(y)\} \quad \text{s.t. set } E(x,y) \text{ of constraints} \tag{8}$$

and

$$x \succsim^P y \ \Leftrightarrow \ D(x,y) \geq 0$$

where :

$$D(x,y) = Max\{U(x) - U(y)\} \quad \text{s.t. set } E(x,y) \text{ of constraints} \tag{9}$$

What are the relations between $\succsim^P$ and $\succsim^N$ with respect to computation? In other words, is it necessary to calculate $d(x,y)$ and $D(x,y)$ for all $x,y \in A$? Proposition 4.5 gives a technical result useful for answering this question.

**Proposition 4.5.** *For all $x, y \in A$, the following equivalences hold:*

$$d(x,y) \geq 0 \Leftrightarrow D(y,x) \leq 0$$
$$D(x,y) \geq 0 \Leftrightarrow d(y,x) \leq 0$$
$$d(x,y) = 0 \Leftrightarrow D(y,x) = 0$$

*Proof.* The proof results from the following equalities:
$d(x,y) = Min_{s.t.E(x,y)}\{U(x)-U(y)\} = -Max_{s.t.E(y,x)}\{U(y)-U(x)\} = -D(y,x)$ □

According to Proposition 4.5, the relation $\succsim^N$ can be computed using either $d(x,y)$ or $D(x,y)$, as shown in Tables 1 and 2. A similar remark concerns the relation $\succsim^P$ which can be computed using either $d(x,y)$ or $D(x,y)$, as shown in Tables 3 and 4.

| | | $y \succsim^N x$ | | not($y \succsim^N x$) |
|---|---|---|---|---|
| | | $d(y,x) > 0$ | $d(y,x) = 0$ | $d(y,x) < 0$ |
| $x \succsim^N y$ | $d(x,y) > 0$ | | | $x \succ^N y$ |
| | $d(x,y) = 0$ | | $x \sim^N y$ | $x \succ^N y$ |
| not($x \succsim^N y$) | $d(x,y) < 0$ | $y \succ^N x$ | $y \succ^N x$ | $x?y$ |

Table 1: Necessary ranking computed in terms of $d(x,y)$

| | | not($x \succsim^N y$) | $x \succsim^N y$ | |
|---|---|---|---|---|
| | | $D(y,x) > 0$ | $D(y,x) = 0$ | $D(y,x) < 0$ |
| not($y \succsim^N x$) | $D(x,y) > 0$ | $x?y$ | $x \succ^N y$ | $x \succ^N y$ |
| $y \succsim^N x$ | $D(x,y) = 0$ | $y \succ^N x$ | $x \sim^N y$ | |
| | $D(x,y) < 0$ | $y \succ^N x$ | | |

Table 2: Necessary ranking computed in terms of $D(x,y)$

| | | $y \succsim^P x$ | | not($y \succsim^P x$) |
|---|---|---|---|---|
| | | $d(y,x) > 0$ | $d(y,x) = 0$ | $d(y,x) < 0$ |
| $x \succsim^P y$ | $d(x,y) > 0$ | | | $x \succ^P y$ |
| | $d(x,y) = 0$ | | $x \sim^P y$ | $x \sim^P y$ |
| not($x \succsim^P y$) | $d(x,y) < 0$ | $y \succ^P x$ | $x \sim^P y$ | $x \sim^P y$ |

Table 3: Possible ranking computed in terms of $d(x,y)$

| | | not($x \succsim^P y$) | $x \succsim^P y$ | |
|---|---|---|---|---|
| | | $D(y,x) > 0$ | $D(y,x) = 0$ | $D(y,x) < 0$ |
| not($y \succsim^P x$) | $D(x,y) > 0$ | $x \sim^P y$ | $x \sim^P y$ | $x \succ^P y$ |
| $y \succsim^P x$ | $D(x,y) = 0$ | $x \sim^P y$ | $x \sim^P y$ | |
| | $D(x,y) < 0$ | $y \succ^P x$ | | |

Table 4: Possible ranking computed in terms of $D(x,y)$

**Remark 4.1.** *In the absence of any pairwise comparison of reference alternatives, the necessary weak preference relation* $\succsim^N$ *boils down to the weak dominance relation* $\Delta$ *in* $A$ $(a\Delta b$ *iff* $g_i(a) \geq g_i(b)$, $i = 1,..,n)$. *Each pairwise comparison provided by the DM, for which the dominance relation does not hold, contributes to enrich* $\succsim^N$*, i.e. it makes the relation* $\succsim^N$ *true for at least one more pair of alternatives.*

**Remark 4.2.** *In the absence of any pairwise comparison of reference alternatives, the possible weak preference relation* $\succsim^P$ *is a complete relation such that for any pair* $(a,b) \in A \times A$

- $a \succsim^P b$ *and* $b \succsim^P a \;\; \Leftrightarrow \;\; ($*not*$(a\Delta b)$ *and not*$(b\Delta a))$ *or* $(a\Delta b$ *and* $b\Delta a)$
- $a \succsim^P b$ *and not*$(b \succsim^P a) \;\; \Leftrightarrow \;\; a\Delta b$ *and not*$(b\Delta a)$

*Each pairwise comparison provided by the DM, for which the dominance relation does not hold, contributes to impoverish* $\succsim^P$*, i.e., it makes the relation* $\succsim^P$ *false for at least one more pair of alternatives.*

## 4.4 Analysis of incompatibility

Let us consider now the case where there is no value function compatible with the preference information. We say, this is the case of incompatibility. In such a case, the polyhedron generated by constraints $E^{A^R}$ is empty. Therefore, the polyhedrons generated by constraints $E(x,y)$, for all $x, y \in A$, are also empty in this case. Such a case may occur in one of the following situations:

- the preferences of the DM do not match the additive model,
- the DM may have made an error in his/her statements; for example stating that $a \succ b$ while $b\Delta a$,

- the statements provided by the DM are contradictory because his/her preferences are unstable, some hidden criteria are taken into account, ...

In such a case, the DM may want either to pursue the analysis with such an incompatibility or to identify its reasons in order to remove it and, therefore, to define a new set of pairwise comparisons $B'^R$ whose corresponding constraints $E'^{A^R}$ generate a non empty polyhedron. Let us consider below the two possible solutions.

### 4.4.1 Situation where incompatibility is accepted

If the DM wants to pursue the analysis with the incompatibility, he/she has to accept that some of his/her pairwise comparisons of reference alternatives will not be reproduced by any value function. Note that, from a formal viewpoint, if the polyhedron generated by $E^{A^R}$ is empty, then $\succsim^N$ and $\succsim^P$ are meaningless. Thus, the acceptance of the inconsistency means that the DM does not change the preference information represented by $\succsim$ and computes $d(x,y)$ and $D(x,y)$ on a new set of constraints $E'^{A^R}$ differing from the original set $E^{A^R}$ by an additional constraint on the acceptable total error:

$$\left.\begin{array}{l}\left.\begin{array}{l}U(a)+\sigma^+(a)-\sigma^-(a)>U(b)+\sigma^+(b)-\sigma^-(b)\ \Leftrightarrow\ a\succ b\\ U(a)+\sigma^+(a)-\sigma^-(a)=U(b)+\sigma^+(b)-\sigma^-(b)\ \Leftrightarrow\ a\sim b\end{array}\right\}\forall a,b\in A^R\\ u_i(g_i(a_{\tau_i(j)}))-u_i(g_i(a_{\tau_i(j-1)}))\geq 0,\ i=1,...,n,\ j=2,...,m\\ u_i(g_i(a_{\tau_i(1)}))\geq 0, u_i(g_i(a_{\tau_i(m)}))\leq u_i(\beta_i),\ i=1,...,n,\\ u_i(\alpha_i)=0,\ i=1,...,n\\ \sum_{i=1}^n u_i(\beta_i)=1,\\ \sigma^+(a)\geq 0,\ \sigma^-(a)\geq 0,\ \forall a\in A^R\\ \sum_{a\in A^R}(\sigma^+(a)+\sigma^-(a))\leq\delta\end{array}\right\}\ (E'^{A^R})$$

where $\delta > F^*$, with $F^* = min\sum_{a\in A^R}(\sigma^+(a)+\sigma^-(a))$ subject to $E^{A^R}$, such that the resulting new set of constraints $E'^{A^R}$ is not empty.

On the basis of $E'^{A^R}$, for any pair $(x,y)\in A$, the set of constraints $E'(x,y)$ can be built as the union of the constraints $E'^{A^R}$ and the constraints relative to the breakpoints introduced by those alternatives $x,y$ that possibly do not belong to $A^R$. Then preference relations $\succsim'^N$ and $\succsim'^P$ can be computed by minimizing and maximizing $U(x)-U(y)$ subject to $E'(x,y)$, rather than to $E(x,y)$, respectively. In other words, in this case, $d(x,y)$ and $D(x,y)$ are computed considering $E'(x,y)$ rather than $E(x,y)$.

Obviously, the necessary and possible rankings resulting from these computations will not fully restore the provided pairwise comparisons, i.e. there is at least one couple $x, y \in A^R$ such that

- $x \succsim y$, but it is false that for all the compatible value functions $U(x) \geq U(y)$ (in other words, there exists a compatible value function such that $U(x) < U(y)$ and thus not($x \succsim'^N y$)), or
- $x \succ y$, but it is false that for all the compatible value functions $U(x) > U(y)$ (in other words, there exists also a value function such that $U(y) \geq U(x)$ and thus $y \succsim'^P x$).

Next result will state that $\succsim'^N$ and $\succsim'^P$ maintain all the main properties of preference relations $\succsim^N$ and $\succsim^P$.

**Proposition 4.6.**

- $\succsim'^N \subseteq \succsim'^P$,
- $\succsim'^N$ *is a complete preorder (i.e. transitive and strongly complete),*
- $\succsim'^P$ *is strongly complete.*

**Proof**: $\succsim'^N$ and $\succsim'^P$ are built using the value functions satisfying constraints $E'^{A^R}$, in the same way as $\succsim^N$ and $\succsim^P$ are built using the value functions satisfying constraints $E^{A^R}$. Thus the proof is analogous to the proof of Propositions 4.1, 4.3 and 4.4. □

### 4.4.2 Situation where incompatibility is not accepted

If the DM does not want to pursue the analysis with the incompatibility, it is necessary to identify the troublesome pairwise comparisons responsible for this incompatibility, so as to remove some of them. Remark that there may exist several sets of pairwise comparisons which, once removed, make set $E^{A^R}$ of constraints non-empty. Hereafter, we outline the main steps of a procedure which identifies these sets.

Recall that the pairwise comparisons of reference alternatives are represented in the ordinal regression constraints $E^{A^R}$ by linear constraints. Hence, identifying the troublesome pairwise comparisons of reference alternatives amounts at finding a minimal subset of constraints that, once removed from $E^{A^R}$, leads to a set of constraints generating a non-empty polyhedron of compatible value functions. The identification procedure is to be performed iteratively since there may exist several minimal subsets of this kind.

Let associate with each pairwise comparison of reference alternatives $a$ and $b$ a new binary variable $v_{a,b}$. Using these binary variables, we rewrite the first two constraints of set $E^{A^R}$ as follows:

$$\begin{array}{l} a \succ b \Leftrightarrow U(a) - U(b) + Mv_{a,b} > 0 \\ a \sim b \Leftrightarrow \left\{ \begin{array}{l} U(a) - U(b) + Mv_{a,b} \geq 0 \\ U(b) - U(a) + Mv_{a,b} \geq 0 \end{array} \right. \end{array} \tag{10}$$

where $M > 1$. Remark that if $v_{a,b} = 1$, then the corresponding constraint is satisfied whatever the value function is, which is equivalent to elimination of this constraint. Therefore, identifying a minimal subset of troublesome pairwise comparisons can be performed by solving the following mixed 0-1 linear program:

$$\begin{array}{ll} Min \rightarrow & f = \sum_{a,b \in A^R:\, a \succsim b} v_{a,b} \\ \text{s. t.} & \\ & \left. \begin{array}{l} a \succ b \Leftrightarrow U(a) - U(b) + Mv_{a,b} \geq \varepsilon \\ a \sim b \Leftrightarrow \left\{ \begin{array}{l} U(a) - U(b) + Mv_{a,b} \geq 0 \\ U(b) - U(a) + Mv_{a,b} \geq 0 \end{array} \right. \end{array} \right\} \forall a, b \in A^R \\ & u_i(g_i(a_{\tau_i(j)})) - u_i(g_i(a_{\tau_i(j-1)})) \geq 0,\ i = 1, ..., n,\ j = 2, ..., m \\ & u_i(g_i(a_{\tau_i(1)})) \geq 0, u_i(g_i(a_{\tau_i(m)})) \leq u_i(\beta_i),\ i = 1, ..., n, \\ & u_i(\alpha_i) = 0,\ i = 1, ..., n \\ & \sum_{i=1}^{n} u_i(\beta_i) = 1, \end{array} \tag{11}$$

The optimal solution of (11) indicates one of the subsets of smallest cardinality being the cause of incompatibility. Alternative subsets of this kind can be found by solving (11) with additional constraint that forbids finding again the same solution. Let $f^*$ be the optimal value of the objective function of (11) and $v^*_{a,b}$ the values of the binary variables at the optimum. Let also $S_1 = \{(a, b) \in A^R \times A^R : a \succsim b$ and $v^*_{a,b} = 1\}$. The additional constraint has then the form

$$\sum_{(a,b) \in S_1} v_{a,b} \leq f^* - 1 \tag{12}$$

Continuing in this way, we can identify other subsets, possibly all of them. These subsets of pairwise comparisons are to be presented to the DM as alternative solutions for removing incompatibility. Such procedure has been described in [20].

# 5 Extensions

## 5.1 Specification of pairwise comparisons with gradual confidence levels

The UTA^GMS method presented in the previous section is intended to support the DM in an interactive process. Indeed, defining a large set of pairwise comparisons of reference alternatives can be difficult for the DM. Therefore, one way to reduce the difficulty of this task would be to permit the DM an incremental specification of pairwise comparisons. This way of proceeding allows the DM to control the evolution of the necessary and possible weak preference relations.

Another way of reducing the difficulty of the task is to extend the UTA^GMS method so as to account for different confidence levels assigned to pairwise comparisons. Let $\succsim_1 \subseteq \succsim_2 \subseteq ... \subseteq \succsim_s$ be embedded sets of DM's partial preorders of reference alternatives. To each set of partial preorders $\succsim_t, t = 1, \ldots s$, corresponds a set of constraints $E_t^{A^R}$ generating a polyhedron of compatible value functions $P_t^{A^R}$. Polyhedrons $P_t^{A^R}$, $t = 1, \ldots s$, are embedded in the inverse order of the related partial preorders $\succsim_t$, i.e. $P_1^{A^R} \supseteq P_2^{A^R} \supseteq ... \supseteq P_s^{A^R}$. We suppose that $P_s^{A^R} \neq \emptyset$ and, therefore, due to the fact that partial preorders $\succsim_t$ are embedded, $P_t^{A^R} \neq \emptyset$, for all $t = 1, \ldots, s$. If $P_s^{A^R} = \emptyset$ we consider only embedded partial preorders until $\succsim_p$ with $p = max\left\{t : P_t^{A^R} \neq \emptyset\right\}$ and relabel $p$ by $s$. For all $x, y \in A$, we say that there is a necessary weak preference relation of level $t$, denoted by $x \succsim_t^N y$ ($t = 1, ..., s$), if for all value functions $U$ compatible with the partial preorder $\succsim_t$, we have $U(x) \geq U(y)$. Analogously, for all $x, y \in A$, we say that there is a possible weak preference relation of level $t$, denoted by $x \succsim_t^P y$ ($t = 1, ..., s$), if for at least one value function $U$ compatible with the partial preorder $\succsim_t$, we have $U(x) \geq U(y)$.

In order to compute possible and necessary weak preference relations $\succsim_t^P$ and $\succsim_t^N$, we can proceed as follows. For all $x, y \in A$, set of constraints $E_t(x, y)$ can be obtained from set $E_t^{A^R}$ by adjoining the constraints relative to the breakpoints introduced by those alternatives $x, y$ that possibly do not belong to $A^R$. For each $t = 1, \ldots, s$, and binary preference relations $\succsim_t^N$ and $\succsim_t^P$, we have

$$x \succsim_t^N y \Leftrightarrow d_t(x, y) \geq 0$$

where:

$$d_t(x, y) = Min\{U(x) - U(y)\} \quad \text{s.t. set } E_t(x, y) \text{ of constraints} \tag{13}$$

and

$$x \succsim_t^P y \;\Leftrightarrow\; D_t(x,y) \geq 0$$

where :
$$D_t(x,y) \;=\; Max\{U(x) - U(y)\} \quad \text{s.t. set } E_t(x,y) \text{ of constraints} \tag{14}$$

Each time we pass from $\succsim_{t-1}$ to $\succsim_t$, $t = 1, \ldots, s-1$, we add to $E_{t-1}^{A^R}$ and, consequently, to $E_{t-1}(x,y)$, new constraints concerning pairs $(a,b) \in A^R \times A^R$, such that $a \succsim_t b$ but not $a \succsim_{t-1} b$, thus the computations of $d_t(x,y)$ and $D_t(x,y)$, for all $x, y \in A \times A$ proceed iteratively.

The following result states that binary preference relations $\succsim_t^N$ and $\succsim_t^P$, $t = 1, \ldots, s$, inherit properties of $\succsim^N$ and $\succsim^P$.

**Proposition 5.1.**

- $\succsim_t^N \subseteq \succsim_t^P$,
- $\succsim_t^N$ *is a complete preorder (i.e. transitive and strongly complete),*
- $\succsim_t^P$ *is strongly complete and negatively transitive.*

*Proof.* Analogous to the proof of Propositions 4.1, 4.3 and 4.4 □

An important property of preference relations $\succsim_t^N$ and $\succsim_t^P$, $t = 1, \ldots, s$ is stated by the following proposition.

**Proposition 5.2.**

$\succsim_t^N$ *and* $\succsim_t^P$, $t = 1, \ldots, s$, *are nested partial preorders:* $\succsim_{t-1}^N \subseteq \succsim_t^N$ *and* $\succsim_t^P \supseteq \succsim_{t-1}^P$, $t = 2, \ldots, s$.

*Proof.* $x \succsim_{t-1}^N y$, $x, y \in A$, means that $U(x) \geq U(y)$ for all value functions $U$ satisfying $E_{t-1}^{A^R}$. Since each value function $U$ satisfying $E_t^{A^R}$ satisfies also $E_{t-1}^{A^R}$, we have that $U(x) \geq U(y)$ for all value functions $U$ satisfying $E_t^{A^R}$, from which we get $x \succsim_t^N y$. Thus $x \succsim_{t-1}^N y \Rightarrow x \succsim_t^N y$, i.e. $\succsim_{t-1}^N \subseteq \succsim_t^N$.
$x \succsim_t^P y$, $x, y \in A$, means that there exists at least one value function $U$ satisfying $E_t^{A^R}$ such that $U(x) \geq U(y)$. Let us denote one of these value functions by $U^*$. Since each value function $U$ satisfying $E_t^{A^R}$ satisfies also $E_{t-1}^{A^R}$, we have that $U^*$ satisfies $E_{t-1}^{A^R}$. Therefore, from $U^*(x) \geq U^*(y)$, we get $x \succsim_{t-1}^P y$. Thus, $x \succsim_t^P y \Rightarrow x \succsim_{t-1}^P y$, i.e. $\succsim_{t-1}^P \supseteq \succsim_t^P$. □

Let $\lambda_t$ be the confidence level assigned to pairwise comparisons concerning pairs $(a,b) \in A^R \times A^R$, such that $a \succsim_t b$ but not $a \succsim_{t-1} b$, $\succsim_0 = \emptyset$, $t = 1,...,s$, $1 = \lambda_1 > \lambda_2 > \ldots > \lambda_s > 0$. Using partial preorders $\succsim_1, \ldots \succsim_s$ and corresponding $\lambda_1, \lambda_2, \ldots, \lambda_s$, a valued binary preference relation $R^N : A \times A \rightarrow [0,1]$ or, more precisely, $R^N : A \times A \rightarrow \{\lambda_1, \lambda_2, \ldots, \lambda_s, 0\}$, can be built as follows: for all $x, y \in A$

- if there exists one $t$ ($t = 1, \ldots, s$) such that $x \succsim_t^N y$, then $R^N(x,y) = max\left\{\lambda_t, t = 1, \ldots, s, \text{such that } x \succsim_t^N y\right\}$
- if there exists no $t$ ($t = 1, \ldots, s$) for which $x \succsim_t^N y$, then $R^N(x,y) = 0$.

Analogously, a valued binary preference relation $R^P : A \times A \rightarrow [0,1]$ or, more precisely, $R^P : A \times A \rightarrow \{1-\lambda_1, 1-\lambda_2, \ldots, 1-\lambda_s, 1\}$, can be built as follows: for all $x, y \in A$

- if there exists one $t$ ($t = 1, \ldots, s$) such that $x \succsim_t^P y$, then $R^P(x,y) = min\left\{1-\lambda_t, t = 1, \ldots, s, \text{such that not } (x \succsim_t^P y)\right\}$
- if $x \succsim_t^P y$ for all $t$ ($t = 1, \ldots, s$), then $R^P(x,y) = 1$.

**Proposition 5.3.** *For all* $x, y \in A$

$$R^N(x,y) = \lambda_{t^*} \Leftrightarrow x \succsim_r^N y \; \forall r \geq t^* \textit{ and not } (x \succsim_r^N y) \; \forall r < t^*;$$

$$R^P(x,y) = 1 - \lambda_{t^*} \Leftrightarrow x \succsim_r^P y \; \forall r < t^* \textit{ and not } (x \succsim_r^P y) \; \forall r \geq t^*.$$

*Proof.* Since $R^N(x,y) = max\left\{\lambda_t, t = 1, \ldots, s, \text{such that } x \succsim_t^N y\right\}$, then $R^N(x,y) = \lambda_{t^*}$ implies $x \succsim_{t^*}^N y$. Taking into account that, for Proposition 5.2, $x \succsim_{t-1}^N y \Rightarrow x \succsim_t^N y$, we have $x \succsim_r^N y \; \forall r \geq t^*$. Moreover, $R^N(x,y) = \lambda_{t^*}$ implies not $(x \succsim_r^N y) \; \forall r$ such that $\lambda_r > \lambda_{t^*}$. Taking into account that $\lambda_t > \lambda_{t+1}$ ($t = 1, ..., s-1$), we get that $R^N(x,y) = \lambda_{t^*}$ implies not $(x \succsim_r^N y) \; \forall r < t^*$. Thus, we proved that

$$R^N(x,y) = \lambda_{t^*} \Rightarrow x \succsim_r^N y \; \forall r \geq t^* \text{ and not} (x \succsim_r^N y) \; \forall r < t^*.$$

For all $x, y \in A$, $x \succsim_r^N y \; \forall r \geq t^*$ and not $(x \succsim_r^N y) \; \forall r < t^* \Rightarrow t^* = min\{t, \; t = 1, \ldots, s, \text{such that } x \succsim_t^N y\}$. $(i)$

Remembering that $\lambda_t > \lambda_{t+1}$ ($t = 1, ..., s-1$), from $(i)$ we get

$$\lambda_{t^*} = max\left\{\lambda_t, t = 1, \ldots, s, \text{such that } x \succsim_t^N y\right\},$$

and for the definition of $R^N(x,y)$, $R^N(x,y) = \lambda_{t^*}$. Thus we proved that

$$x \succsim_r^N y \; \forall r \geq t^* \text{ and not } (x \succsim_r^N y) \; \forall r < t^* \Rightarrow R^N(x,y) = \lambda_{t^*},$$

which concludes the proof of

$$R^N(x,y) = \lambda_{t^*} \Leftrightarrow x \succsim_r^N y \ \forall r \geq t^* \text{ and not}(x \succsim_r^N y) \ \forall r < t^*.$$

Analogous proof holds for

$$R^P(x,y) = 1 - \lambda_{t^*} \Leftrightarrow x \succsim_r^P y \ \forall r < t^* \text{ and not } (x \succsim_r^P y) \ \forall r \geq t^*.$$

□

Proposition 5.3 says that $R^N(x,y) = \lambda_t$ means that $x \succsim_r^N y$ holds only for $r \geq t$, while, for definition, $R^N(x,y) = 0$ means that $x \succsim_t^N y$ does not hold for any $t$ ($t = 1,\ldots,s$). Proposition 5.3 says also that $R^P(x,y) = 1 - \lambda_t$ means that $x \succsim_r^P y$ hols only for $r < t$, while, for definition, $R^P(x,y) = 1$ means that $x \succsim_t^P y$ for all $t$ ($t = 1,\ldots,s$).

It is interesting to investigate the properties of valued binary relations $R^N$ and $R^P$ (for an introduction to valued binary relations and their properties see [6]). Let us remind that a valued binary relation $R$ defined on a set $X$, i.e. $R : X \times X \rightarrow [0,1]$, is

- reflexive, if for all $x \in X$, $R(x,x) = 1$,
- min-transitive, if for all $x,y,z \in X$, $min(R(x,y), R(y,z)) \leq R(x,z)$,
- strongly complete, if for all $x,y \in X$, $max(R(x,y), R(y,x)) = 1$,
- negatively transitive, if for all $x,y,z \in X$, $min((1 - R(x,y)), (1 - R(y,z))) \leq (1 - R(x,z))$.

A valued binary relation which is reflexive and min-transitive is called fuzzy partial preorder.

**Proposition 5.4.** *Valued binary relation $R^N$ is reflexive and min-transitive and, therefore, it is a fuzzy partial preorder. Valued binary relation $R^P$ is strongly complete and negatively transitive.*

*Proof.* For all $x \in A$, for all value functions $U$ compatible with the partial preorder $\succsim_s$, we have $U(x) = U(x)$, which implies $x \succsim_s^N x$ and $R^N(x,x) = 1$, i.e. $R^N$ is reflexive.

For all $x,y,z \in X$, two cases are possible:

a) $min(R^N(x,y), R^N(y,z)) = 0$,

b) $min(R^N(x,y), R^N(y,z)) > 0$.

Considering that always $R^N(x,z) \geq 0$, in case a),

$$R^N(x,z) \geq min(R^N(x,y), R^N(y,z)). \quad (i)$$

In case b), for the definition of $R^N$, we have that

$$min(R^N(x,y), R^N(y,z)) =$$

$$= min\left\{max\left\{\lambda_t, t = 1, \ldots, s, \text{such that } x \succsim_t^N y\right\}, max\left\{\lambda_t, t = 1, \ldots, s, \text{such that } y \succsim_t^N z\right\}\right\} =$$

$$= max\left\{\lambda_t, t = 1, \ldots, s, \text{such that } x \succsim_t^N y \text{ and } y \succsim_t^N z\right\}.$$

Thus, if $min(R^N(x,y), R^N(y,z)) = \lambda_r$, then $U(x) \geq U(y)$ and $U(y) \geq U(z)$ for all value functions $U$ compatible with $\succsim_r$. Thus, for all value functions $U$ compatible with $\succsim_r$ we have $U(x) \geq U(z)$ and, consequently, $x \succsim_r^N z$. This implies that

$$\begin{aligned} R^N(x,z) &= max\left\{\lambda_t, t = 1, \ldots, s, \text{such that } x \succsim_t^N z\right\} \\ &\geq \lambda_r = min(R^N(x,y), R^N(y,z)). \quad (ii) \end{aligned}$$

For $(i)$ and $(ii)$, valued binary relation $R^N$ is min-transitive.

Let us suppose that $x \succsim_s^P y$, $x, y \in A$. In this case, for Proposition 5.2, $x \succsim_t^P y$ for all $t$ ($t = 1, \ldots, s$) and, therefore, $R^P(x,y) = 1$. If, instead, not ($x \succsim_s^P y$), then for Proposition 4.2, $y \succsim_s^N x$ and, therefore, for Property 1, $y \succsim_s^P x$. In consequence, for Proposition 5.2, $y \succsim_t^P x$ for all $t$ ($t = 1, \ldots, s$) and, thus, $R^P(y,x) = 1$. This proves completeness of valued binary relation $R^P$.

For all $x, y, z \in X$, two cases are possible:

a) $max(R^P(x,y), R^P(y,z)) = 1$,

b) $max(R^N(x,y), R^N(y,z)) < 1$.

In case $a)$ we have that $R^P(x,y) = 1$ or $R^P(y,z) = 1$, and thus $1 - R^P(x,y) = 0$ or $1 - R^P(y,z) = 0$, such that $min((1 - R^P(x,y)), (1 - R^P(y,z))) = 0$ and considering that always$1 - R^P(x,z) \geq 0$, we get

$$min((1 - R^P(x,y)), (1 - R^P(y,z))) \leq 1 - R^P(x,z).$$

In case $b)$, $R^P(x,y) < 1$ and $R^P(y,z) < 1$, for definition of $R^P$

,

$$1 - R^P(x,y) = 1 - min\left\{1 - \lambda_t, t = 1, \ldots, s, \text{such that not } (x \succsim_t^P y)\right\}) =$$

$$= max\left\{\lambda_t, t = 1, \ldots, s, \text{such that not } (x \succsim_t^P y)\right\})$$

as well as

$$1 - R^P(y,z) = 1 - min\left\{1 - \lambda_t, t = 1, \ldots, s, \text{such that not } (y \succsim_t^P z)\right\}) =$$

$$= max\left\{\lambda_t, t = 1, \ldots, s, \text{such that not } (y \succsim_t^P z)\right\}).$$

Thus,

$$min((1 - R^P(x,y)), (1 - R^P(y,z)))$$

$$=$$

$$min\left\{max\left\{\lambda_t, t = 1, \ldots, s, \text{such that not } (x \succsim_t^P y)\right\}, \right.$$
$$\left. max\left\{\lambda_t, t = 1, \ldots, s, \text{such that not } (y \succsim_t^P z)\right\}\right\}$$

$$=$$

$$max\left\{\lambda_t, t = 1, \ldots, s, \text{such that not } (x \succsim_t^P y) \text{ and not } (y \succsim_t^P z)\right\},$$

$$=$$

$$max\{\lambda_t, t = 1, \ldots, s, \text{such that, } U(y) > U(x) \text{ and } U(z) > U(y)$$

$$\text{for all value functions } U \text{ compatible with } \succsim_t\}.$$

If $min((1 - R^P(x,y)), (1 - R^P(y,z))) = \lambda_r$, then $U(x) < U(z)$ for all value functions compatible with $\succsim_r$, which means that there does not exist any value function $U$ compatible with $\succsim_r$ such that $U(x) \geq U(z)$. Thus, $min\{t, t = 1, \ldots, s,$ such that not $(x \succsim_t^P z)\} \leq r$ and, therefore,

$$max\{\lambda_t, t = 1, \ldots, s, \text{ such that not } (x \succsim_t^P z)\} \geq \lambda_r.$$

Since

$$max\{\lambda_t, t = 1, \ldots, s, \text{ such that not } (x \succsim_t^P z)\} =$$

$$= 1 - min\{1 - \lambda_t, t = 1, \ldots, s, \text{such that not} (x \succsim_t^P z)\} = 1 - R^P(x,z),$$

we conclude that $(1 - R^P(x,z)) \geq \lambda_r = min((1 - R^P(x,y)), (1 - R^P(y,z)))$.

□

### 5.2 Accounting for intensity of preference

Another preference information that can be provided by the DM concerns the intensity of preference among two pairs of reference alternatives. Given two pairs of alternatives $(x, y) \in \succsim$ and $(w, z) \in \succsim$, such that $x \succ y$ and $w \succ z$, the DM can state : "$x$ *is preferred to* $y$ *at least as much as* $w$ *is preferred to* $z$". Such statement means that for all compatible value functions $U$:

$$U(x) - U(y) > U(w) - U(z). \quad (15)$$

To account for the above preference information, it is sufficient to include condition (15) in set $E^{A^R}$ of constraints. Of course, consequently, condition (15) will be included in constraints $E(x, y)$ for all $x, y \in A$.

Conversely, $\forall x, y, w, z \in A$, it is possible to check whether or not condition

$$U(x) - U(y) > U(w) - U(z) \quad (16)$$

holds for all compatible value functions $U$.

In order to do so, it is sufficient to check the feasibility of constraints $E(x, y)$ and (16). Such information may enrich the DM's knowledge of his/her preferences.

## 6 Illustrative example

In this section, we illustrate how a decision aiding process can be supported by the $\text{UTA}^{\text{GMS}}$ method. The computations of this example were performed using the VISUAL-UTA software (see [30]). We consider the following hypothetical decision problem. AGRITEC is a medium size firm (350 persons approx.) producing some tools for agriculture. The C.E.O., M$^{\text{r}}$ Becault, intends to double the production and multiply exports by 4 within 5 years. Therefore, he wants to hire a new international sales manager. A recruitment agency has interviewed 17 potential candidates which have been evaluated on 3 criteria (sales management experience, international experience, human qualities) evaluated on a [0,100] scale. The evaluations of candidates are provided in Table 5. Without any further information, the computed partial preorder $\succsim_0^N$ corresponds to the weak dominance relation $\Delta$ on the set of alternatives (See Figure 2).

The C.E.O. has attended 4 interviews and can express a confident judgement about theses candidates: Ferret and Frechet are equally good, Fourny is less acceptable than Ferret and Frechet, and Fleichman is even less acceptable than Fourny. This means that

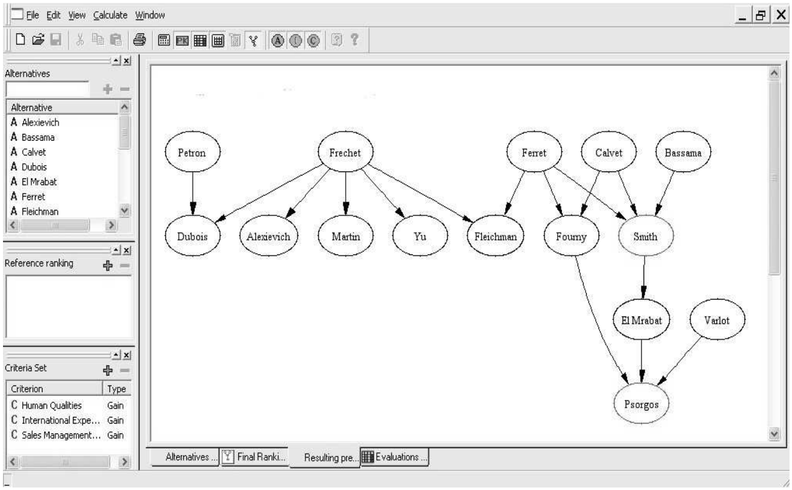



Figure 2: Partial preorder $\succsim_0^N$ corresponding to the weak dominance relation $\Delta$

the initial reference ranking is the following: Ferret $\sim$ Frechet $\succ$ Fourny $\succ$ Fleichman. For this initial preference information, the partial preorder $\succsim_1^N$ has been computed using UTA$^{\text{GMS}}$ (See Figure 3).

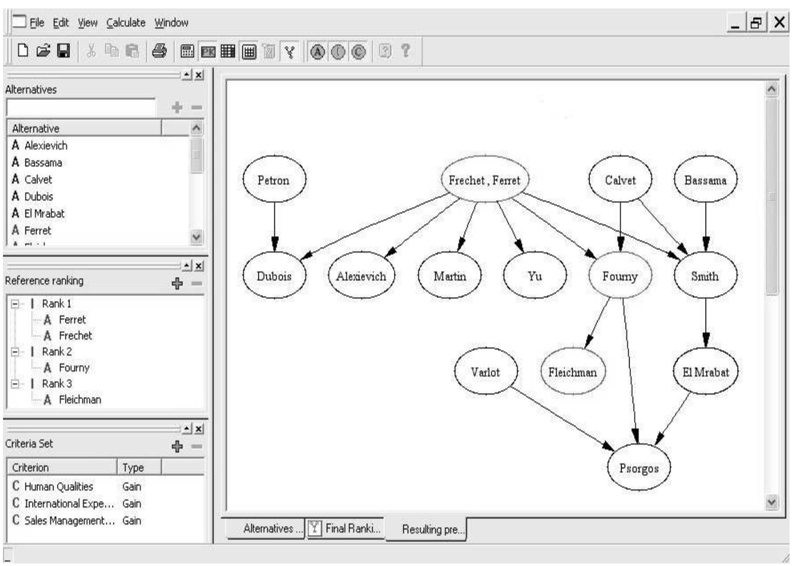



Figure 3: Partial preorder $\succsim_1^N$

| | Crit 1 | Crit 2 | Crit 3 |
|---|---|---|---|
| Alexievich | 4 | 16 | 63 |
| Bassama | 28 | 18 | 28 |
| Calvet | 26 | 40 | 44 |
| Dubois | 2 | 2 | 68 |
| El Mrabat | 18 | 17 | 14 |
| Feeret | 35 | 62 | 25 |
| Fleichman | 7 | 55 | 12 |
| Fourny | 25 | 30 | 12 |
| Frechet | 9 | 62 | 88 |
| Martin | 0 | 24 | 73 |
| Petron | 6 | 15 | 100 |
| Psorgos | 16 | 9 | 0 |
| Smith | 26 | 17 | 17 |
| Varlot | 62 | 43 | 0 |
| Yu | 1 | 32 | 64 |

Table 5: Evaluation Table

Considering this first result, Mr Becault is willing to add further preference information. This results in the following new reference ranking: Ferret $\sim$ Frechet $\succ$ Martin $\succ$ Fourny $\sim$ El Mrabat $\succ$ Fleichman. However, as he did not attend the interview of El Mrabat and Martin, his opinion about the relative ranking of these candidates is not as certain as the initial preference information.

It appears that for the provided information, no additive value function fits the last reference ranking. The analysis of this incompatibility reveals that the statement Ferret $\sim$ Frechet cannot be represented together with the statement Fourny $\sim$ El Mrabat by an additive value function. In other words, it is necessary for Mr Becault to revise one of these statements. As he did not interview El Mrabat, he decides to remove him from the reference ranking which becomes Ferret $\sim$ Frechet $\succ$ Martin $\succ$ Fourny $\succ$ Fleichman. This reference ranking is compatible with a representation by an additive value function. Figure 4 represents two nested partial preorders:

- bold arrows represent partial preorder $\succsim_1^N$ obtained for the most certain preference information only, i.e., Ferret $\sim$ Frechet $\succ$ Fourny $\succ$ Fleichman,
- dashed arrows represent partial preorder $\succsim_2^N$ obtained for the consistent preference information composed of the most certain preference information and the less confident preference information about Martin, i.e., Ferret $\sim$ Frechet $\succ$ Martin $\succ$ Fourny

$\succ$ Fleichman.

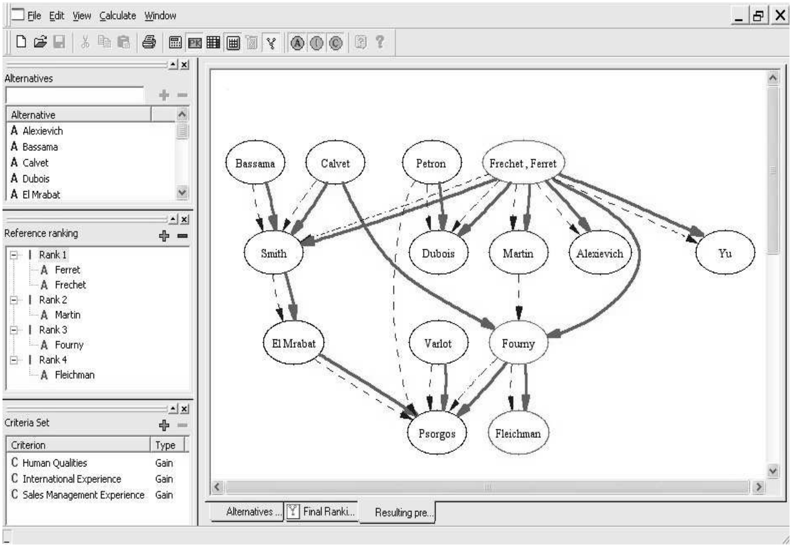



Figure 4: Nested partial preorders $\succsim_2^N$ (bold) and $\succsim_3^N$ (dashed)

The interactive process can be pursued, M[r] Becault adding in iteration $t$ some new pairwise comparisons of reference alternatives, thus enriching the resulting partial preorder $\succsim_t^N$, until it is decisive enough for the C.E.O. to make his choice.

# 7 Conclusion

The new UTA$^{GMS}$ method presented in this paper is an ordinal regression method supporting multiple criteria ranking of alternatives; it is distinguished from previous methods of this kind by the following new features:

- the method considers general additive value functions rather than piecewise linear ones,
- the final rankings are defined using all value functions compatible with the provided preference information,
- the method provides two final rankings: the necessary ranking identifies “sure” preference statements while the possible ranking identifies “possible” preference statements,

- distinguishing necessary and possible consequences of using all value functions compatible with preference information, UTA$^{GMS}$ includes a kind of robustness analysis instead of using a single "best-fit" value function,
- the necessary and possible preference relations considered in UTA$^{GMS}$ have several properties of general interest for MCDA,
- when the DM provides preference information that cannot be represented by an additive model, the method identifies which pieces of the information underly this impossibility,
- the method does not require the DM to interpret (and even look at) the marginal value functions,
- the DM can assign confidence levels to pieces of preference information, which yields a valued necessary preference (proved to be a fuzzy partial preorder) and a valued possible preference (proved to be a strongly complete and negatively transitive valued binary relation).

We envisage the following future developments of the presented methodology:

- application to multicriteria sorting problems,
- application to group decision problems,
- application to interactive multiobjective optimization.

# References

[1] C.A. Bana e Costa and J.C.. Vansnick. MACBETH - an interactive path towards the construction of cardinal value fonctions. *International transactions in operational Research*, 1:489–500, 1994.

[2] Barron, F.H. Selecting a best multiattribute alternative with partial information about attribute weights *IActa Psychologica*, 80:91–103, 1992.

[3] V. Belton and T. Stewart. *Multiple Criteria Decision Analysis: An integrated approach.* Kluwer Academic Publishers, 2002.

[4] J.M. Devaud, G. Groussaud, and E. Jacquet-Lagreze. UTADIS: Une méthode de construction de fonctions d'utilité additives rendant compte de jugements globaux. In *Proceedings of the European working group on MCDA, Bochum , Germany*, 1980.

[5] L. C. Dias and J. N. Clímaco. Additive Aggregation with Variable Interdependent Parameters: the VIP Analysis Software. *Journal of the Operational Research Society*, 51:1070-1082, 2000.

[6] J. Fodor and M. Roubens. *Fuzzy preference modelling and multicriteria decision support*. Kluwer Academic Publishers, 1994.

[7] S. Greco, B. Matarazzo, and R. Slowinski. The use of rough sets and fuzzy sets in MCDM. In T. Gal, T. Stewart, and T. Hanne, editors, *Advances in MCDM models, Algorithms, Theory, and Applications*, pages 14.1–14.59. Kluwer Academic, Dordrecht, 1999.

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

[9] S. Greco., B. Matarazzo, and R. Slowinski. Decision rule approach. In J. Figueira, S. Greco, M. Ehrgott, editors, *Multiple Criteria Decision Analysis, State of the Art Surveys*, chapter 13, pages 507–561. Springer, New York, 2005.

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

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

[12] C.W. Kirkwood and R.K. Sarin. Ranking with partial information: A method and an application. *Operations Research*, 33(1):38–48, January-February 1985.

[13] L. Kiss, J.M. Martel, and R. Nadeau. ELECCALC - an interactive software for modelling the decision maker's preferences. *Decision Support Systems*, 12(4-5):757–777, November 1994.

[14] M. Koksalan and V. Taner. An approach for finding the most preferred alternative in the presence of multiple criteria. *European Journal of Operational Research*, 60(1):53–60, 1992.

[15] M. Koksalan and C. Ulu. An interactive approach for placing alternatives in preference classes. *European Journal of Operational Research*, 144:429–439, 2003.

[16] R. Lahdelma, J. Hokkanen and P. Salminen, SMAA - Stochastic multiobjective acceptability analysis, *European Journal of Operational Research* , 106:137-143, 1998.

[17] K.S. Lee, K.S. Park, and S.H. Kim. Dominance, potential optimality, imprecise information, and hierarchical structure in multi-criteria analysis. *Computers and Operations Research*, 29:1267–1281, 2002.

[18] J.G. March. Bounded rationality, ambiguity and the engineering of choice. *Bell Journal of Economics*, 9:587–608, 1978.

[19] R.S. Michalski, I. Bratko, and M. Kubat (*eds.*). *Machine learning and datamining - Methods and applications*. Wiley, New York, 1998.

[20] V. Mousseau, J. Figueira, L.C. Dias, C. Gomes da Silva, and J.N. Clímaco. Resolving inconsistencies among constraints on the parameters of an MCDA model. *European Journal of Operational Research*, 147(1):72–93, May 2003.

[21] V. Mousseau and R. Slowinski. Inferring an ELECTRE TRI model from assignment examples. *Journal of Global Optimization*, 12(2):157–174, 1998.

[22] D. Pekelman and S.K. Sen. Mathematical programming models for the determination of attribute weights. *Management Science*, 20(8):1217–1229, April 1974.

[23] B. Roy. *Multicriteria Methodology for Decision Aiding*. Kluwer Academic, Dordrecht, 1996.

[24] B. Roy and D. Bouyssou. *Aide Multicritère à la Décision: Méthodes et Cas*. Economica, Paris, 1993.

[25] T.L. Saaty. The Analytic Hierarchy and Analytic Network Processes for the Measurement of Intangible Criteria and for Decision-Making. in J. Figueira and S. Greco and M. Ehrgott (ed.) *IEEE Multiple Criteria Decision Analysis: State of the Art Surveys*, 345-408, Springer Verlag, 2005.

[26] A.A. Salo and R.P. Hamalainen. Preference ratio in multiattribute evaluation (PRIME) - elicitation and decision procedures under incomplete information. *IEEE Transactions on Systems, Man and Cybertnetics: Part A*, 31(6):533–545, November 2001.

[27] Y. Siskos. A way to deal with fuzzy preferences in multicriteria decision problems. *European Journal of Operational Research*, 10(3):314–324, 1982.

[28] Y. Siskos, E. Grigoroudis, and N.F. Matsatsinis. UTA Methods. In J. Figueira, S. Greco, M. Ehrgott, editors, *State of the Art in Multiple Criteria Decision Analysis*. Springer, 2005.

[29] V. Srinivasan and A.D. Shocker. Estimating the weights for multiple attributes in a composite criterion using pairwise judgments. *Psychometrika*, 38(4):473–493, 1973.

[30] VisualUTA Software, v.1.0, 2006, Laboratory of Intelligent Decision Support Systems, Institute of Computing Science, Poznan University of Technology, available at `http://idss.cs.put.poznan.pl/site/software.html`.

[31] M. Weber. A method of multiattribute decision making with incomplete information. *Management Science*, 31(11):1365–1371, 1985.

# L'$\alpha$-robustesse lexicographique : une relaxation de la $\beta$-robustesse

Rim Kalaï*, Claude Lamboray†

**Résumé**

Dans ce papier, nous nous intéressons à deux approches de robustesse : d'un côté une approche fondée sur la quasi-optimalité que nous appellerons dans cet article la $\beta$-robustesse et d'un autre côté l'$\alpha$-robustesse lexicographique. Nous mettons en évidence quelques propriétés communes aux deux approches. En outre, nous établissons un lien entre la $\beta$-robustesse et l'$\alpha$-robustesse lexicographique qui montre que la seconde approche peut être considérée comme une relaxation de la première.

**Mots-clefs :** $\alpha$-robustesse lexicographique, $\beta$-robustesse.

**Abstract**

In this paper, we are interested in two particular robustness approaches: on the one hand an approach based on the concept of quasi-optimality which we will call $\beta$-robustness and on the other hand lexicographic $\alpha$-robustness. We will focus on properties that these two approaches have in common. Furthermore, we will establish a link between lexicographic $\alpha$-robustness and $\beta$-robustness which will show that the first approach is indeed a relaxation of the latter approach.

**Key words :** Lexicographic $\alpha$-robustness, $\beta$-robustness.

*LAMSADE, Université Paris-Dauphine, 75775 Paris cedex 16, France / ESC Rouen, 76825 Mont-Saint-Aignan Cedex, France. {kalai@lamsade.dauphine.fr}

†SMG, Université Libre de Bruxelles, B-1050 Bruxelles, Belgique. clambora@ulb.ac.be

# 1 Introduction

Pour répondre à la préoccupation de robustesse, des approches extrêmement variées peuvent être envisagées. Nous nous intéressons ici à une forme de réponse particulièrement étudiée dans la littérature actuelle illustrée notamment dans [3, 4, 8, 12, 14, 15, 17]. Ces approches se divisent en deux grandes familles : celles qui se basent sur l'optimisation d'un critère de robustesse et celles qui imposent des conditions de robustesse qu'une solution doit vérifier pour être considérée comme robuste [2]. Dans ce travail, nous nous intéressons à deux approches de robustesse appartenant à la seconde famille. Nous nous limitons au cas où l'incertitude est représentée par un ensemble discret et fini de scénarios.

Dans la première approche, une solution est dite robuste si elle n'est pas très loin de l'optimum quel que soit le scénario considéré. Cette mesure de robustesse a été initialement introduite par Kouvelis et al. dans le cadre du problème d'aménagement d'usine [11]. Les auteurs cherchent à déterminer les solutions dont le coût est à moins de $(100 + p)\%$ de celui de la solution optimale pour tous les scénarios, $p$ étant un réel positif fixé. Les solutions robustes sont donc celles dont le regret relatif est inférieur à $p\%$. Dans [19], Snyder appelle cette mesure de robustesse la *p-robustesse*. Ce concept de robustesse basé sur la proximité de l'optimum a été également repris par Gutiérrez et al. dans le cas du problème de conception de réseaux (*network design problem*) [6] et du problème d'approvisionnement international (*international sourcing*) [5], ainsi que par Mulvey et al. en programmation mathématique [14].

La seconde approche se fonde sur le fait qu'en situation d'incertitude pure, aucun scénario ne peut être distingué et donc l'ordre des scénarios n'a pas d'importance. Remarquons que l'ordre des scénarios n'a pas d'importance non plus dans la première approche. Néanmoins, la seconde approche va davantage plus loin en supposant que le vecteur de performance d'une solution (représentant les valeurs possibles que prend la solution pour les différents scénarios) est équivalent à tout vecteur obtenu par permutation, notamment le vecteur obtenu en ordonnant les performances de la pire à la meilleure. Une solution est alors considérée comme robuste, selon cette seconde approche, si chacune de ses performances réordonnées est proche de l'optimum. Cette approche, appelée *$\alpha$-robustesse lexicographique*, a été proposée par Kalaï dans [7] et a été appliquée au problème de localisation 1-médian [8] ainsi qu'aux problèmes de sac à dos [9] et de plus court chemin [10].

En plus du fait d'appartenir à la famille d'approches imposant des conditions de robustesse, plusieurs points communs existent entre ces deux approches :

- Elles cherchent un ensemble de solutions et non pas une seule solution robuste.
- Elles peuvent donner un ensemble vide de solutions robustes, à la différence des approches fondées sur l'optimisation d'un critère de robustesse, comme par exemple le critère du coût ou du regret maximal.
- Elles tiennent compte de la dimension subjective de la robustesse [20] par le biais

d'un seuil d'indifférence à fixer par le décideur. En effet, ce paramètre traduit la perception du décideur d'être "proche de l'optimum".
- Elles considèrent tous les scénarios pour déterminer les solutions robustes, à la différence de l'approche minmax qui ne tient compte que du pire scénario.

Le but de ce travail est de mettre en relation ces deux approches ayant des philosophies similaires. Par souci d'homogénéité et vu que la $p$-robustesse fait appel à un seuil relatif alors que la $\alpha$-robustesse lexicographique utilise un seuil absolu, nous proposons de comparer cette dernière avec l'approche qui considère qu'une solution est robuste si son regret absolu (non relatif) est inférieur à un seuil $\beta$ pour tous les scénarios. Nous appelons cette dernière approche la *$\beta$-robustesse*. Soulignons que la $\beta$-robustesse et la $p$-robustesse reflètent le même concept de robustesse, la seule différence réside dans le choix de la nature du seuil (absolu ou relatif).

Le reste de l'article est organisé comme suit. Dans la section 2, nous introduisons les notations utilisées le long de cet article ainsi qu'un exemple illustratif. La section 3 est dédiée à certaines propriétés vérifiées par les deux approches. Un lien entre la $\beta$-robustesse et la $\alpha$-robustesse lexicographique est établi dans la section 4. La section 5 est consacrée aux conclusions.

# 2 Notations et exemple

## 2.1 Notations

Nous considérons un ensemble fini de solutions ou d'alternatives, $X = \{x_1, x_2, \ldots, x_n\}$ et un nombre fini de scénarios $S$, avec $|X| = n$ et $|S| = q$. Chaque alternative $x \in X$ est évaluée pour le scénario $s$ par une fonction $f_s(x)$ (par exemple le coût). Sans perte de généralité, nous allons supposer que les fonctions $f_s(.)$ sont à minimiser.

Notons $f(x)$ le vecteur d'évaluations $(f_{s^1}(x), f_{s^2}(x), \ldots, f_{s^q}(x))$ de la solution $x$ pour les différents scénarios. Le *vecteur de désutilité* de $x$, noté $\hat{f}(x)$, est obtenu en réordonnant $f(x)$ de la pire (c'est-à-dire la plus grande) évaluation à la meilleure (c'est-à-dire la plus petite) évaluation. On a donc :

$$\hat{f}_1(x) \geq \hat{f}_2(x) \geq \ldots \geq \hat{f}_q(x)$$

Notons que le vecteur de désutilité d'une solution $x$ introduit un préordre complet sur l'ensemble des scénarios qui dépend de la solution $x$ considérée. Soit $x_s^*$ une solution optimale pour le scénario $s$ (elle peut ne pas être unique) :

$$x_s^* = \arg\min_{x \in X} f_s(x) \quad (1)$$

De façon similaire, $\hat{x}_j^*$ est défini par rapport à la fonction $\hat{f}_j(.)$ comme suit :

$$\hat{x}_j^* = \arg\min_{x \in X} \hat{f}_j(x) \tag{2}$$

A partir de ces solutions, nous définissons deux nouvelles solutions notées $x^*$ et $\hat{x}^*$ vérifiant :

$$f_s(x^*) = f_s(x_s^*) \qquad \forall s \in S \tag{3}$$

$$\hat{f}_j(\hat{x}^*) = \hat{f}_j(\hat{x}_j^*) \quad \forall j \in \{1, \ldots, q\} \tag{4}$$

Si $f$ est une fonction coût par exemple, la solution $x^*$ correspond à la solution dont le coût est minimum pour chacun des $q$ scénarios, alors que la solution $\hat{x}^*$ correspond à la solution dont le $k^{\text{ème}}$ plus grand coût est minimum pour tout $k \in \{1, \ldots, q\}$. Signalons que, la plupart du temps, $x^*$ et $\hat{x}^*$ ne sont pas des solutions réalisables mais correspondent à des solutions idéales. Elles jouent le rôle de *points de référence* auxquelles nous comparerons les solutions de $X$ pour déterminer celles qui peuvent être considérées comme robustes.

Soient $\alpha$ et $\beta$ deux réels positifs. Dans ce qui suit, nous allons noter :

- $\mathcal{R}_\beta$ l'ensemble des *solutions $\beta$-robustes* défini comme suit :

$$\mathcal{R}_\beta(X) \;=\; \{x \in X : \forall s \in S,\ f_s(x) - f_s(x_s^*) \leq \beta\} \tag{5}$$

- $\mathcal{R}_{\alpha-lex}$ l'ensemble des solutions *$\alpha$-robustes lexicographiques* défini comme suit :

$$\mathcal{R}_{\alpha-lex}(X) \;=\; \{x \in X : \forall j \in \{1, \ldots, q\},\ \hat{f}_j(x) - \hat{f}_j(\hat{x}_j^*) \leq \alpha\} \tag{6}$$

Dans la sous-section qui suit, nous illustrons ces deux ensembles par un exemple pour que le lecteur puisse mieux se les représenter.

## 2.2 Exemple

Considérons l'exemple indiqué dans le tableau 1 avec $X = \{a_1, a_2, a_3, a_4\}$ et $S = \{s_1, s_2\}$ :

Le point idéal $x^*$ de ce jeu de données correspond à la solution qui prend les valeurs 1 dans le premier scénario et 3 dans le deuxième scénario. Cette solution ainsi que celles de l'ensemble $X$ sont représentées dans la Figure 1, où les axes correspondent aux deux scénarios. Pour une valeur de $\beta = 4$ par exemple, l'ensemble robuste $\mathcal{R}_\beta(X)$ est constitué de l'unique solution $a_2$. En effet, $a_2$ est à moins de 4 unités du point de référence à la fois pour le premier et le second scénario.

Afin de calculer l'ensemble $\mathcal{R}_{\alpha-lex}(X)$, les vecteurs de performance des différentes solutions doivent être réordonnés comme le montre le Tableau 2.

| | $s_1$ | $s_2$ |
|---|---|---|
| $f_s(a_1)$ | 6 | 3 |
| $f_s(a_2)$ | 4 | 7 |
| $f_s(a_3)$ | 3 | 10 |
| $f_s(a_4)$ | 1 | 11 |
| $f_s(x^*)$ | 1 | 3 |

TAB. 1 – Evaluations des quatre solutions pour les deux scénarios

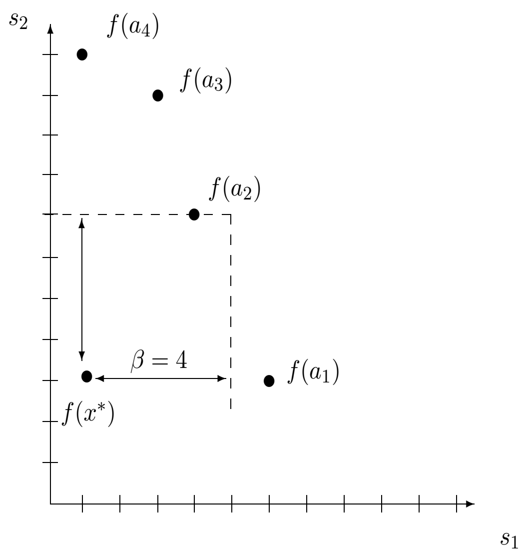



FIG. 1 – Evaluation des solutions dans les deux scénarios.

| | $j=1$ | $j=2$ |
|---|---|---|
| $\hat{f}_j(a_1)$ | 6 | 3 |
| $\hat{f}_j(a_2)$ | 7 | 4 |
| $\hat{f}_j(a_3)$ | 10 | 3 |
| $\hat{f}_j(a_4)$ | 11 | 1 |
| $\hat{f}_j(\hat{x}^*)$ | 6 | 1 |

TAB. 2 – Evaluations réordonnées des quatre solutions

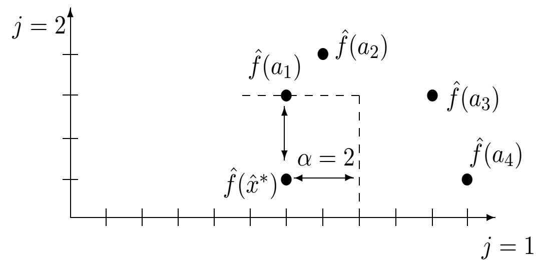



FIG. 2 – Evaluations réordonnées des solutions.

La solution $\hat{x}^*$ correspond maintenant au point idéal du jeu de données réordonné. Remarquons que les nouvelles dimensions ne correspondent plus forcément à un scénario en particulier. Par exemple, $\hat{f}_1(a_1) = 6 = f_1(a_1)$ provient du premier scénario et $\hat{f}_1(a_2) = 7 = f_2(a_2)$ provient du deuxième scénario. Ces solutions sont représentées dans la Figure 2. Pour une valeur $\alpha = 2$ par exemple, l'ensemble robuste $\mathcal{R}_{\alpha-lex}(X)$ est constitué de l'unique solution $a_1$. En effet, $a_1$ est à moins de 2 unités du point idéal $\hat{x}^*$ à la fois pour la première et la deuxième composantes.

Les Figures 1 et 2 montrent une propriété intéressante des deux approches étudiées dans cet article et évoquée dans l'introduction. Pour une valeur $\beta < 4$ (resp. $\alpha$ < 2), l'ensemble $\mathcal{R}_\beta(X)$ (resp. $\mathcal{R}_{\alpha-lex}(X)$) est vide. En effet, les seuils $\beta$ et $\alpha$ expriment un niveau de tolérance fixé par le décideur pour exprimer son point de vue quant à la robustesse d'une solution par rapport au point idéal. S'il fixe un seuil trop faible et si le point de référence n'est pas une solution réalisable, il n'y aura pas de solutions robustes selon ses exigences. Nous expliciterons, dans la section 3 les seuils minimaux garantissant l'existence de solutions robustes pour les deux approches, ainsi que plusieurs autres propriétés.

# 3 Propriétés des ensembles $\mathcal{R}_\beta(X)$ et $\mathcal{R}_{\alpha-lex}(X)$

Dans cette section, nous mettons en évidence certaines propriétés vérifiées par les deux ensembles de solutions robustes $\mathcal{R}_{\alpha-lex}(X)$ et $\mathcal{R}_\beta(X)$. Pour se référer génériquement aux deux ensembles, nous les notons $\mathcal{R}$ ou encore $\mathcal{R}(Y)$ pour spécifier qu'ils sont définis sur un sous-ensemble $Y \subseteq X$.

**Propriété 1** *Soit $a \in X$. Si $f_s(a) \leq f_s(x)$ pour tout $x \in X$ et $s \in S$, alors $a \in \mathcal{R}(X)$.*

**Preuve:**
*Remarque préliminaire* : Soient $x$ et $y$ deux solutions de $X$. Podinovskii démontre dans

[16] que $\hat{f}_j(x) \leq \hat{f}_j(y)$ pour tout $j$ si et seulement si il existe deux permutations $\sigma$ et $\pi$ telles que $f_{\sigma(s)}(x) \leq f_{\pi(s)}(y)$ pour tout $s$. En particulier, si $f_s(x) \leq f_s(y)$ pour tout $s \in S$ alors $\forall j \in \{1,\ldots,q\}, \hat{f}_j(x) \leq \hat{f}_j(y)$.

Par définition de $a$, $f_s(a) = f_s(x_s^*)$ pour tout $s \in S$. Donc $a \in \mathcal{R}_\beta(X)$. Par ailleurs et d'après le résultat de Podinovskii, on a $\forall k \leq q, \hat{f}_k(a) \leq \hat{f}_k(x)$ pour tout $x \in X$. Donc, $\forall k, \hat{f}_k(a) = \hat{f}_k(\hat{x}_k^*)$ et $a \in \mathcal{R}_{\alpha-lex}(X)$. □

On comprend bien que la propriété 1 est une propriété souhaitée dans tout ensemble de solutions robustes. En effet, s'il existe dans $X$ une solution qui domine toutes les autres, c'est-à-dire une même solution qui est optimale dans tous les scénarios, il est évident que cette solution doit être considérée comme robuste. Dans ce cas particulier, les ensembles $\mathcal{R}_\beta(X)$ et $\mathcal{R}_{\alpha-lex}(X)$ sont non vides quel que soit le seuil choisi.

**Propriété 2** *Les ensembles $\mathcal{R}_\beta(X)$ et $\mathcal{R}_{\alpha-lex}(X)$ sont monotones respectivement par rapport à $\beta$ et à $\alpha$, i.e. :*

$$\begin{aligned} \beta \leq \beta' &\Rightarrow \mathcal{R}_\beta(X) \subseteq \mathcal{R}_{\beta'}(X) \\ \alpha \leq \alpha' &\Rightarrow \mathcal{R}_{\alpha-lex}(X) \subseteq \mathcal{R}_{\alpha'-lex}(X) \end{aligned}$$

**Preuve:** $x \in \mathcal{R}_\beta(X) \Rightarrow \forall s \in S, f_s(x) - f_s(x_s^*) \leq \beta \leq \beta' \Rightarrow x \in \mathcal{R}_{\beta'}(X)$.
$x \in \mathcal{R}_{\alpha-lex}(X) \Rightarrow \forall k \leq q, \hat{f}_k(x) - \hat{f}_k(\hat{x}_k^*) \leq \alpha \leq \alpha' \Rightarrow x \in \mathcal{R}_{\alpha'-lex}(X)$. □

Les paramètres $\beta$ et $\alpha$ caractérisent le degré d'exigence du décideur par rapport à une solution idéale. Il est clair que plus cette exigence est faible (et donc plus le paramètre est grand), plus il y a de solutions qui répondent aux préférences du décideur.

Il peut arriver que, pour des seuils $\beta$ ou $\alpha$ trop petits, aucune solution dans $X$ ne vérifie la condition de robustesse requise. Par conséquent, les ensembles $\mathcal{R}_\beta(X)$ et $\mathcal{R}_{\alpha-lex}(X)$ peuvent être vides. Notons $\beta_{min}$ le seuil minimum pour lequel $\mathcal{R}_\beta(X)$ est non vide et $\alpha_{min}$ le seuil minimum pour lequel $\mathcal{R}_{\alpha-lex}(X)$ est non vide. Soulignons que ces seuils minimaux dépendent de l'ensemble $X$. En toute rigueur, ils doivent être notés $\beta_{min}(X)$ et $\alpha_{min}(X)$. Néanmoins, par souci de simplicité, nous n'utiliserons ces dernières notations que s'il y a possibilité de confusion. Il est évident que, suite à la propriété de monotonie (propriété 2), les ensembles $\mathcal{R}_\beta(X)$ et $\mathcal{R}_{\alpha-lex}(X)$ restent non vides pour tout $\beta \geq \beta_{min}$ et tout $\alpha \geq \alpha_{min}$.

**Proposition 1**

$$\beta_{min} = \min_{x \in X} \max_{s \in S} (f_s(x) - f_s(x_s^*))$$

$$\alpha_{min} = \min_{x \in X} \max_{1 \leq j \leq q} (\hat{f}_j(x) - \hat{f}_j(\hat{x}_j^*))$$

**Preuve:** $x \in \mathcal{R}_\beta(X) \Leftrightarrow \forall s \in S, f_s(x) - f_s(x_s^*) \leq \beta \Leftrightarrow \max_{s\in S}(f_s(x) - f_s(x_s^*)) \leq \beta$
Donc, $\beta_{min} = \min_{x\in X} \max_{s\in S}(f_s(x) - f_s(x_s^*))$.
Pour $\alpha_{min}$, même preuve que précédemment en remplaçant $f_s(.)$ par $\hat{f}_j(.)$, $x_s^*$ par $\hat{x}_j^*$ et $\beta$ par $\alpha$. □

De manière similaire, il existe un seuil à partir duquel chaque solution est robuste. Remarquons que, d'après la proposition 1, l'ensemble $\mathcal{R}_{\beta_{min}}(X)$ correspond exactement à l'ensemble des solutions optimales suivant le critère du regret maximal.

**Propriété 3** *Les ensembles $\mathcal{R}_\beta(X)$ et $\mathcal{R}_{\alpha-lex}(X)$ vérifient la propriété suivante :*

$$\textit{Si } Y \subseteq X \textit{ alors } \big(\mathcal{R}(X) \cap Y\big) \subseteq \mathcal{R}(Y)$$

**Preuve:** Soient $Y$ une partie de $X$ et $a$ une solution telle que $a \in \mathcal{R}_\beta(X) \cap Y$.
On a donc $f_s(a) - f_s(x_s^*) \leq \beta$, $\forall s \in S$. Posons $x_s^*(Y) = \arg\min_{x\in Y} f_s(x)$. Par définition de $x_s^*$, on a $f_s(x_s^*(Y)) \geq f_s(x_s^*)$. Donc, pour tout $s \in S$, $f_s(a) - f_s(x_s^*(Y)) \leq \beta$, ce qui signifie $a \in \mathcal{R}_\beta(Y)$. Idem pour $\mathcal{R}_{\alpha-lex}(X)$. □

Cette propriété est appelée *propriété de Chernoff* [13] ou encore *condition d'hérédité* [1] en théorie de choix social. Elle signifie qu'une solution robuste sur un ensemble reste robuste sur une partie de cet ensemble. En d'autres termes, si une alternative n'est pas considérée comme robuste par rapport à une partie de $X$, elle ne peut être considérée robuste par rapport à l'ensemble $X$.

En particulier, si $a \notin \mathcal{R}(X)$, on peut alors déduire de cette propriété que $\mathcal{R}(X) \subseteq \mathcal{R}(X\backslash\{a\})$ (il suffit de considérer $Y = X\backslash\{a\}$ dans la propriété 3). Il est donc intéressant de remarquer que, lorsqu'une solution de $X$ n'appartenant pas à l'ensemble robuste est enlevée, l'ensemble robuste risque de s'agrandir. Soulignons que cela ne se produit que si l'on enlève une solution $x$ telle qu'il existe $s$ pour lequel $f_s(x) = f_s(x_s^*)$, respectivement il existe $j$ pour lequel $\hat{f}_j(x) = \hat{f}_j(\hat{x}_j^*)$. En agissant de la sorte, on modifie le point de référence. Comme les ensembles sont définis par rapport à ces points de référence, la modification de ces points influence l'ensemble des solutions robustes.

Pour résumer, si on réduit l'ensemble de solutions, une solution robuste reste robuste (si elle n'est pas enlevée) et une solution non robuste peut devenir robuste. Si, en revanche, on augmente l'ensemble de solutions, une solution initialement non robuste reste non robuste tandis qu'une solution robuste peut cesser de l'être. Cela peut même se produire lorsque l'ensemble des solutions est enrichi d'une seule solution qui ne sera pas considérée comme robuste. Imaginons que, dans l'exemple de la section 2, la solution $f(a_5) = (8, 2)$ est ajoutée. Dès lors, l'ensemble des solutions $\beta$-robustes devient vide pour $\beta = 4$ alors qu'il contenait initialement la solution $a_2$.

D'autre part, si dans la propriété de Chernoff, on considère le cas particulier $Y = \mathcal{R}(X)$, il en découle le résultat suivant :

**Propriété 4** *$\mathcal{R}_{\alpha-lex}(X)$ et $\mathcal{R}_{\beta}(X)$ sont idempotents, i.e. :*

$$\mathcal{R}(\mathcal{R}(X)) = \mathcal{R}(X)$$

L'idempotence des ensembles $\mathcal{R}_{\beta}(X)$ et $\mathcal{R}_{\alpha-lex}(X)$ implique qu'ils sont minimaux en un certain sens, puisqu'en ré-appliquant la démarche de recherche de solutions robustes à ces ensembles, on ne peut pas les réduire.

Les ensembles de solutions $\beta$-robustes et $\alpha$-robustes lexicographiques vérifient une autre propriété faisant référence à deux sous-ensembles de $X$.

**Propriété 5** *Les ensembles $\mathcal{R}_{\beta}(X)$ et $\mathcal{R}_{\alpha-lex}(X)$ vérifient la propriété suivante :*

$$\forall Y_1, Y_2 \subseteq X, \quad \mathcal{R}(Y_1) \cap \mathcal{R}(Y_2) \subseteq \mathcal{R}(Y_1 \cup Y_2)$$

**Preuve:** $a \in \mathcal{R}_{\beta}(Y_1) \cap \mathcal{R}_{\beta}(Y_2) \Rightarrow \begin{cases} f_s(a) - \min_{y_1 \in Y_1} f_s(y_1) \leq \beta & \forall s \in S \\ f_s(a) - \min_{y_2 \in Y_2} f_s(y_2) \leq \beta & \forall s \in S \end{cases}$
Comme $\min_{z \in Y_1 \cup Y_2} f_s(z) = \min\{\min_{y_1 \in Y_1} f_s(y_1), \min_{y_2 \in Y_2} f_s(y_2)\}$, on en déduit que $f_s(a) - \min_{z \in Y_1 \cup Y_2} f_s(z) \leq \beta, \ \forall s \in S$. Ce qui implique $a \in \mathcal{R}(Y_1 \cup Y_2)$.
La preuve est similaire pour $R_{\alpha-lex}(X)$. □

Cette propriété est appelée *propriété d'expansion* [13]. Elle signifie que si une solution est robuste sur deux sous-ensembles séparément, alors elle reste robuste sur l'union de ces deux sous-ensembles.

En théorie du choix social, les propriétés de Chernoff et d'expansion sont importantes car elles caractérisent les fonctions de choix *rationalisables* par une relation ayant une partie asymétrique acyclique (voir théorème de Sen [13, 18])[1]. Une *fonction de choix* est une fonction qui, à tout sous-ensemble non vide $Y \subseteq X$, associe un sous-ensemble non vide $C(Y) \subseteq Y$. Une fonction de choix est dite *rationalisable* lorsqu'il existe une relation binaire $S$ telle que l'ensemble des éléments maximaux de cette relation correspond à l'ensemble de choix. En fait, d'après Moulin [13], cette relation binaire correspond à la *relation de base* $S$ associée à la fonction de choix et définie comme suit :

$$xSy \text{ ssi } x \in C(\{x, y\}) \tag{7}$$

[1]Pour que cette relation acyclique soit quasi-transitive, il faut qu'elle vérifie, en plus des propriétés de Chernoff et d'expansion, une troisième propriété appelée propriété d'Aizermann. Cette dernière stipule que si $\mathcal{R}(X) \subseteq Y \subseteq X$ alors $\mathcal{R}(Y) \subseteq \mathcal{R}(X)$. La propriété d'Aizermann n'est vérifiée ni pour $\mathcal{R}_{\alpha-lex}$ ni pour $\mathcal{R}_{\beta}$, comme le montre le contre-exemple suivant : f(x)=(6,3,3), f(y)=(4,4,4) et f(z)=(5,2,2).

où $C(\{x, y\})$ indique la fonction de choix appliquée au sous-ensemble $\{x, y\}$.

Notons que les ensembles $\mathcal{R}_\beta(X)$ et $\mathcal{R}_{\alpha-lex}(X)$ ne correspondent pas toujours à des fonctions de choix, même s'ils ne sont pas vides sur $X$. En effet, considérons l'exemple de la section 2.2 pour $Y = \{a_1, a_4\}$ et $\beta = \beta_{min}(X) = 4$. On a $\mathcal{R}_\beta(Y) = \emptyset$ alors que $\mathcal{R}_\beta(X) = \{a_2\}$. La même constatation peut être faite sur l'exemple pour l'ensemble $\mathcal{R}_{\alpha-lex}(X)$ en prenant $Y = \{a_2, a_4\}$ et $\alpha = \alpha_{min}(X) = 2$.

Néanmoins, il existe un seuil à partir duquel chacun de ces deux ensembles robustes correspond à une fonction de choix. Notons $\beta^c$ le seuil tel que :

$$\forall Y \subseteq X,\ Y \neq \emptyset \ \Rightarrow\ \mathcal{R}_{\beta^c}(Y) \neq \emptyset \tag{8}$$

De même, notons $\alpha^c$ le seuil tel que :

$$\forall Y \subseteq X,\ Y \neq \emptyset \ \Rightarrow\ \mathcal{R}_{\alpha^c-lex}(Y) \neq \emptyset \tag{9}$$

Il est clair que $\beta^c \geq \beta_{min}(X)$ et $\alpha^c \geq \alpha_{min}(X)$. Dans l'exemple de la section 2.2, on a $\beta^c = 5 > \beta_{min}(X)$ et $\alpha^c = 3 > \alpha_{min}(X)$. D'après la définition de $\beta^c$ et $\alpha^c$, nous avons :

**Proposition 2**

$$\beta^c = \max_{Y \subseteq X, Y \neq \emptyset} \beta_{min}(Y)$$

$$\alpha^c = \max_{Y \subseteq X, Y \neq \emptyset} \alpha_{min}(Y)$$

**Preuve:** On a $\forall Y \subseteq X,\ Y \neq \emptyset,\quad \beta \geq \beta^c \geq \beta_{min}(Y) \ \Rightarrow\ \mathcal{R}_\beta(Y) \neq \emptyset$. Ce qui démontre que $\forall \beta \geq \beta^c$, $\mathcal{R}_\beta(X)$ est une fonction de choix. Il est clair que $\beta^c$ est le plus petit seuil vérifiant cette propriété. Idem pour $\alpha^c$. □

Si $\beta \geq \beta^c$ et $\alpha \geq \alpha^c$, alors les ensembles $\mathcal{R}_\beta(X)$ et $\mathcal{R}_{\alpha-lex}(X)$ sont des fonctions de choix rationalisables par les relations de base suivantes :

$$\begin{aligned}
xS_\beta y \ &\Leftrightarrow\ x \in \mathcal{R}_\beta(\{x, y\}) && (10)\\
&\Leftrightarrow\ f_s(x) - \min\{f_s(x), f_s(y)\} \leq \beta, \forall s \in S && (11)\\
&\Leftrightarrow\ f_s(x) - f_s(y) \leq \beta, \forall s \in S && (12)
\end{aligned}$$

$$\begin{aligned}
xS_\alpha y \ &\Leftrightarrow\ x \in \mathcal{R}_{\alpha-lex}(\{x, y\}) && (13)\\
&\Leftrightarrow\ \hat{f}_j(x) - \min\{\hat{f}_j(x), \hat{f}_j(y)\} \leq \alpha, \forall j \in \{1, \ldots, q\} && (14)\\
&\Leftrightarrow\ \hat{f}_j(x) - \hat{f}_j(y) \leq \alpha, \forall j \in \{1, \ldots, q\} && (15)
\end{aligned}$$

Notons qu'en plus d'avoir une partie asymétrique acyclique, les relations $S_\beta$ et $S_\alpha$ sont complètes [13]. Les ensembles $\mathcal{R}_\beta(X)$ et $\mathcal{R}_{\alpha-lex}(X)$ correspondent respectivement aux éléments maximaux de $S_\beta$ et $S_\alpha$.

**Proposition 3**

$$\forall \beta \geq \beta^c, \mathcal{R}_\beta(X) = \{x \in X : \forall y \in X \;\; xS_\beta y\}$$

$$\forall \alpha \geq \alpha^c, \mathcal{R}_{\alpha-lex}(X) = \{x \in X : \forall y \in X \;\; xS_\alpha y\}$$

Une interprétation de ce résultat est qu'au-delà d'un certain seuil, une solution robuste ne dépend plus du contexte (l'ensemble $X$ entier) mais uniquement des évaluations par paire. En effet, une solution peut être éliminée dès lors qu'elle n'est pas robuste par rapport à une autre.

En plus des points communs énumérés dans l'introduction, nous venons de voir que les deux ensembles de solutions robustes se comportent de façon similaire par rapport à certaines propriétés. Cette ressemblance entre la $\beta$-robustesse et l'$\alpha$-robustesse lexicographique suggère de chercher un lien entre ces deux approches, ce que nous nous proposons de faire dans la section qui suit.

# 4 Liens entre $\mathcal{R}_\beta(X)$ et $\mathcal{R}_{\alpha-lex}(X)$

Dans cette section, nous établissons une relation entre l'ensemble des solutions $\beta$-robustes $\mathcal{R}_\beta(X)$ et l'ensemble des solutions $\alpha$-robustes lexicographiques $\mathcal{R}_{\alpha-lex}(X)$.

Pour donner une idée intuitive du résultat, regardons l'évolution des ensembles $\mathcal{R}_\beta(X)$ et $\mathcal{R}_{\alpha-lex}(X)$ en fonction des seuils $\beta$ et $\alpha$ dans l'exemple de la section 2.2. Le Tableau 3 représente cette évolution pour des seuils variant de 1 à 8. Par exemple, pour $\beta = \alpha = 4$, $\mathcal{R}_\beta(X) = \{a_2\}$ et $\mathcal{R}_{\alpha-lex}(X) = \{a_1, a_2, a_3\}$. Dans cet exemple, il apparaît que pour des seuils $\beta$ et $\alpha$ égaux, l'ensemble $\mathcal{R}_\beta(X)$ est inclus dans l'ensemble $\mathcal{R}_{\alpha-lex}(X)$ quelle que soit la valeur de ces paramètres. Par monotonie de l'ensemble des solutions $\beta$-robustes, cette constatation reste valable pour $\beta \leq \alpha$.

| *Valeur du seuil* | $\mathcal{R}_\beta(X)$ | $\mathcal{R}_{\alpha-lex}(X)$ |
|---|---|---|
| 0 - 1 | $\emptyset$ | $\emptyset$ |
| 2 | $\emptyset$ | $\{a_1\}$ |
| 3 | $\emptyset$ | $\{a_1, a_2\}$ |
| 4 | $\{a_2\}$ | $\{a_1, a_2, a_3\}$ |
| 5 - 6 | $\{a_1, a_2\}$ | $\{a_1, a_2, a_3, a_4\} = X$ |
| 7 | $\{a_1, a_2, a_3\}$ | $\{a_1, a_2, a_3, a_4\} = X$ |
| 8 | $\{a_1, a_2, a_3, a_4\} = X$ | $\{a_1, a_2, a_3, a_4\} = X$ |

TAB. 3 – Les ensembles $\mathcal{R}_\beta(X)$ and $\mathcal{R}_{\alpha-lex}(X)$ en fonction du seuil

Nous démontrons dans ce qui suit que cette relation est toujours vérifiée.

**Théorème 1** *Si* $\beta \leq \alpha$ *alors* $\mathcal{R}_\beta(X) \subseteq \mathcal{R}_{\alpha-lex}(X)$.

**Preuve:** Soit $x \in \mathcal{R}_\beta(X)$. On a :

$$\forall s \in S,\ f_s(x) \leq f_s(x^*) + \beta \tag{16}$$

La solution $x^*$ étant celle définie par l'équation (3). Grâce au résultat de Podinovskii énoncé dans la preuve de la propriété 1, nous pouvons déduire que :

$$\forall j \leq q,\ \hat{f}_j(x) \ \ \leq \ \ \hat{f}_j(x^*) + \beta. \tag{17}$$

Nous allons maintenant montrer que, pour tout $k \leq q$ :

$$\hat{f}_k(x^*) \ \ \leq \ \ \hat{f}_k(\hat{x}^*_k). \tag{18}$$

Par définition de $x^*$, nous avons, pour tout $x \in X$ :

$$\forall s \in S :\ f_s(x^*) \leq f_s(x). \tag{19}$$

Cette inégalité est vraie en particulier pour $x = \hat{x}^*_k = \arg\min_{x \in X} \hat{f}_k(x)$, $k$ étant fixé :

$$\forall s \in S :\ f_s(x^*) \leq f_s(\hat{x}^*_k). \tag{20}$$

Nous pouvons déduire que :

$$\forall j \leq q :\ \hat{f}_j(x^*) \leq \hat{f}_j(\hat{x}^*_k). \tag{21}$$

Ceci étant vrai pour tout $j$, c'est vrai pour $j = k$, ce qui démontre l'inégalité (18). Etant données l'inégalité (17) et le fait que $\beta \leq \alpha$, il s'en suit que :

$$\forall j \leq q,\ \hat{f}_j(x) \leq \hat{f}_j(\hat{x}^*_j) + \alpha. \tag{22}$$

Par conséquent, $x \in \mathcal{R}_{\alpha-lex}(X)$. □

Remarquons que la démonstration reste vraie si on considère non pas l'ensemble $X$ entier, mais seulement un sous-ensemble non vide $Y \subseteq X$. Ainsi, si $\beta \leq \alpha$ alors $\mathcal{R}_\beta(Y) \subseteq \mathcal{R}_{\alpha-lex}(Y)$.

Il découle du théorème 1 que les seuils $\alpha_{min}$ et $\beta_{min}$ vérifient la relation suivante :

**Corollaire 1** $\alpha_{min} \leq \beta_{min}$

**Preuve:** Supposons que $\alpha_{min} > \beta_{min}$ et choisissons $\beta = \alpha = \beta_{min}$. On a donc $\mathcal{R}_\beta(X) \neq \emptyset$ et $\mathcal{R}_{\alpha-lex}(X) = \emptyset$ ce qui contredit le théorème 1. □

Les seuils $\beta^c$ et $\alpha^c$, définis respectivement par les équations (8) et (9), vérifient ce qui suit :

**Corollaire 2** $\alpha^c \leq \beta^c$

**Preuve:** Supposons que $\beta = \alpha = \beta^c$. On a donc :

$$\forall Y \subseteq X, Y \neq \emptyset \quad \mathcal{R}_\beta(X) \neq \emptyset$$

D'après le théorème 1, ceci implique que :

$$\forall Y \subseteq X, Y \neq \emptyset \quad \mathcal{R}_{\alpha-lex}(X) \neq \emptyset$$

Par conséquent, $\alpha^c \leq \alpha = \beta^c$. □

Le théorème 1 indique que l'$\alpha$-robustesse lexicographique, tout en ayant beaucoup de propriétés communes avec la $\beta$-robustesse, est moins "sévère" que cette dernière puisqu'elle donne des solutions pour des seuils plus faibles que ceux exigés par la seconde approche. Cette caractéristique semble donner un argument en défaveur de l'$\alpha$-robustesse lexicographique puisque, généralement, on cherche à réduire un ensemble de choix et non à l'augmenter. Néanmoins, rappelons que le seuil $\beta_{min}$, donnant le plus petit ensemble de solutions $\beta$-robustes non vide, n'est autre que le regret maximal minimal. Cette valeur peut être importante dans certains cas ; dans l'exemple de la section 2.2, $\beta_{min}$ est deux fois plus grand que $\alpha_{min}$. Il n'est donc pas toujours possible de trouver des solutions $\beta$-robustes pour des seuils raisonnables, d'où l'intérêt d'avoir une approche similaire qui donne des solutions pour des seuils plus faibles.

# 5 Conclusion

Dans cet article, nous avons voulu mettre en évidence la similitude entre l'$\alpha$-robustesse lexicographique et la $\beta$-robustesse sur la base d'un certain nombre de propriétés. De plus, nous avons pu établir un lien direct entre ces deux approches qui suggère que la première approche est en quelque sorte une relaxation de la seconde.

Ce travail de comparaison entre les deux approches ne constitue qu'un premier pas dans cette direction. Nous nous interrogeons, par exemple, sur les problèmes pour lesquels les deux seuils minimaux $\alpha_{min}$ et $\beta_{min}$ sont différents. La réponse à cette question pourrait donner plus de crédibilité à l'$\alpha$-robustesse lexicographique si de tels problèmes étaient nombreux. Il faudrait aussi étudier davantage la relation de base qui rationalise les deux ensembles, dans un but algorithmique par exemple. Finalement, une caractérisation axiomatique des deux ensembles de solutions robustes permettra de pouvoir clairement les distinguer.

De manière plus générale, nous pensons qu'il est utile d'élargir ce type de comparaison à d'autres approches de robustesse existantes. En effet, cela permettra de mieux guider un analyste ou un décideur dans le choix de l'approche robuste la plus appropriée dans une application concrète.

# Références


[1] ALESKEROV F. and MONJARDET B. *Utility maximization, choice and preference*. Springer, 2002.

[2] ALOULOU M.A., KALAÏ R., and VANDERPOOTEN D. Une nouvelle approche de robustesse : $\alpha$-robustesse lexicographique. *Bulletin du Groupe de Travail Européen Aide Multicritère à la Décision*, Series 3(12), automne 2005.

[3] BERTSIMAS D. and SIM M. Robust discrete optimization and network flows. *Mathematical Programming*, Ser.B 98 :49–71, 2003.

[4] GUPTA S.K. and ROSENHEAD J. Robustness in sequential investment decisions. *Management Science*, 15(2) :B18–B29, october 1968.

[5] GUTIÉRREZ G.J. and KOUVELIS P. A robustness approach to international sourcing. *Annals of Operations Research*, 59 :165–193, 1995.

[6] GUTIÉRREZ G.J., KOUVELIS P., and KURAWARWALA S.A. A robustness approach to uncapacitated network design problem. *European Journal of Operational Research*, 94(2) :362–376, 1996.

[7] KALAÏ R. *Une nouvelle approche de robustesse : application à quelques problèmes d'optimisation*. Phd thesis, LAMSADE, Université Paris-Dauphine, Paris, France, october 2006.

[8] KALAÏ R., ALOULOU M.A., VALLIN P., and VANDERPOOTEN D. Robust 1-median location problem on a tree. In *Proceedings of the third ORP3 conference (a Euro Conference for Young OR Reserchers and Practitioners)*, pages 201–212, Valencia, Spain, 6-10 september 2005.

[9] KALAÏ R. and VANDERPOOTEN D. Lexicographic $\alpha$-robust knapsack problem : Complexity results. In *International Conference on Service Systems & Service Management (IEEE SSSM'06)*, Troyes, France, 25-27 october 2006.

[10] KALAÏ R. and VANDERPOOTEN D. Lexicographic $\alpha$-robust shortest path problem. In *International Network Optimization Conference (INOC 2007)*, Spa, Belgique, april 2007.

[11] KOUVELIS P., KURAWARWALA A.A., and GUTIÉRREZ G.J. Algorithms for robust single and multiple period layout planning for manufacturing systems. *European Journal of Operational Research*, 63(2) :287–303, 1992.

[12] KOUVELIS P. and YU G. *Robust Discrete Optimization and its Applications*. Kluwer Academic Publishers, 1997.

[13] MOULIN H. Choice functions over a finite set : a summary. *Social Choice Welfare*, 2 :147–160, 1985.

[14] MULVEY J.M., VANDERBEI R.J., and ZENIOS S.A. Robust optimization of large-scale systems. *Operations Research*, 43(2) :264–281, 1995.

[15] PERNY P. and SPANJAARD O. An axiomatic approach to robustness in search problems with multiple scenarios. *Nineteenth Conference on Uncertainty in Artificial Intelligence*, 8-10 Août 2003. UAI 2003, Acapulco (Mexique).

[16] PODINOVSKII V.V. Multi-criterion problems with uniform equivalent criteria. *USSR Computational Mathematics and Mathematical Physics*, 15 :47–60, 1975.

[17] ROSENBLATT M.J. and LEE H.L. A robustness approach to facilities design. *International Journal of Production Research*, 25(4) :479–486, 1987.

[18] SEN A.K. Choice functions and revealed preferences. *The Review of Economic Studies*, 38(3) :307–317, 1971.

[19] SNYDER L.V. Facility location under uncertainty : A review. *IIE Transactions*, 38(7) :537–554, 2006.

[20] VINCKE P. About robustness analysis. In *Newsletters of the European Working Group «Multicriteria Aid for Decisions»*, number 8 in 3, pages 7–9, 2003.

# Supporting the search for a group ranking with robust conclusions on prudent orders

Claude Lamboray

**Résumé**

Nous analysons le problème où des rangements, fournis par exemple par des évaluateurs, doivent être agrégés en un rangement de compromis. Dans ce contexte, Arrow et Raynaud suggèrent que le rangement de compromis soit un ordre prudent. En général, un ordre prudent n'est pas unique. C'est pourquoi nous proposons de construire des conclusions robustes sur ces ordres prudents afin de gérer la possible multiplicité des solutions de compromis. Cette approche va permettre de progressivement raffiner le modèle d'aide à la décision. Nous illustrons notre approche sur un exemple de rangement de priorités de recherche par un groupe de jeunes chercheurs.

**Mots-clefs :** Décisions de groupe, Rangement de compromis, Ordres prudents

**Abstract**

We consider the problem where rankings, provided for instance by a group of evaluators, have to be combined into a common group ranking. In such a context, Arrow and Raynaud suggested that the compromise ranking should be a prudent order. In general, a prudent order is not unique. That is why, we propose to manage this possible multiplicity of compromise solutions by computing robust conclusions. This allows for a progressive refinement of the decision model and supports the group to eventually select one group ranking. The approach is illustrated on a problem where a group of junior researchers has to agree on a ranking of research domains.

**Key words :** Group decisions, Compromise ranking, Prudent orders

*CoDE, Université Libre de Bruxelles, C.P. 210/01, Boulevard du triomphe, B-1050 Bruxelles, Belgique. clambora@ulb.ac.be, Université du Luxembourg, 162 A, avenue de la Faïencerie, L-1511 Luxembourg, claude.lamboray@uni.lu

# 1 Introduction

Often decision problems that arise in companies or organizations involve a group of individuals. In particular we will focus on the problem where the judgment of different experts, evaluators or decision makers has to be taken into account in order to rank a set of alternatives from the best to the worst. This is typically the case in situations where a group of experts has to agree on a priority list. In the example that we will present at the end of this paper, a group of junior researchers was asked by a scientific funding agency to rank a set of research domains that should be developed in the medium and long term.

Although the group members all pursue a global goal, that is to take a good decision for their company or their organization, they can have a different opinion about how to rank the alternatives. In fact, they may have a different perspective on the problem or they may play a different role within the organization. Inevitably, conflicting preferences will emerge. In such a case, a group decision then consists in reducing the individual preferences into a collective preference (see Jelassi [21]).

This problem be approached in a variety of ways, ranging from a simple ranking solely based on, let's say, cost to complex multicriteria group ranking techniques. De Keyser [13] differentiates between four types of multicriteria multidecision maker models. The last of these four types consists of an approach where, first, each group member explicitly states his individual ranking. In a second stage, mechanisms and tools are then needed that aggregate or combine this multi-dimensional ordinal data into a new, so-called *compromise* or *group* ranking. In this paper we are going to adopt such an approach. As argued by De Keyser [13], this consequently leaves a large amount of freedom to each group member on how to perform his evaluation and construct his personal ranking.

Many ordinal ranking rules have been proposed since more than 200 years. Borda [6] is one of the first who examined the problem. His rule consists in ranking the alternatives according to the sum of ranks. Some other rules have been proposed where the score of each alternative naturally defines a weak order, such as for instance the rules proposed by Copeland [10] or Simpson [30]. The Maximize agreement heuristic proposed by Beck and Lin [2] is slightly different since it follows a ranking by choosing scheme. From a different perspective, many ranking rules have been modeled as optimization problems. The compromise ranking is a linear order that optimizes a certain objective function. The most known such rule has been proposed by Kemeny [22], but let us also mention the rules proposed by Slater [31], Bernardo [3], Blin [5] and Cook and Seiford [9].

In this paper, we will present an approach to support the search for a compromise ranking using the concept of prudent orders. In fact, when working in an "industrial" or "business-like" situation, then Arrow and Raynaud [1] suggested that the compromise ranking should be a prudent order. Intuitively, a prudent order is a compromise ranking such that the strongest opposition against this solution is minimal.

Throughout the paper, we will assume that the input data will consist of linear orders. This can be criticized, especially from an operational point of view. However, this assumption will simplify the underlying aggregation problem. This paper can also be seen as a first step toward designing a similar decision aid tool which would allow for more complex preference structures.

Prudent orders will be presented in section 2. The methodological framework of our approach will be motivated more in depth in section 3. In section 4, we will compute robust conclusions on these prudent orders. We will discuss an adaptation step in section 5. The concepts introduced are illustrated on an example in section 6. We end this paper with a conclusion. All the proofs can be found in the appendix.

# 2 Prudent orders

Let us suppose that a set of alternative has been clearly established and that this set will remain stable during the whole decision process. We will assume that each group member will propose a linear order (antisymmetric, complete and transitive binary relation). These linear orders have then to be aggregated into a group ranking.

Formally, let $\mathcal{O}$ denote the set of all linear orders over a finite set of $n$ alternatives $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$. Let $u = (O_1, O_2, \ldots O_q) \in \mathcal{O}^q$ be a profile of $q$ linear orders of the $q$ group members. A valued relation $S$ counts the number of rankings that prefer $a_i$ over $a_j$:

$$S_{ij} = |\{k \in \{1, \ldots, q\} : (a_i, a_j) \in O_k\}|$$

Since the relations $O_k$ are linear orders, it is easy to see that for $i \neq j$ we have $S_{ij} + S_{ji} = q$. This is called the constant sum property.

Let $\lambda \in \{0, \ldots, q\}$. Let $R_{\geq\lambda}$ ($R_{>\lambda}$) be a cut-relation of $S$ defined as follows: $S_{ij} \geq \lambda \Rightarrow (a_i, a_j) \in R_{\geq\lambda}$ and $S_{ij} > \lambda \Rightarrow (a_i, a_j) \in R_{>\lambda}$. When $\lambda = 0$, $R_{\geq\lambda}$ contains all the ordered pairs. By increasing the value of $\lambda$, less pairs of alternatives will belong to the cut-relation. At one point, it will no longer be possible that $R_{\geq\lambda}$ contains a linear order. In particular we are interested in the largest possible value of $\lambda$ such that $R_{\geq\lambda}$ still contains a linear order.

$$\alpha = \max\{\lambda \in \{0, \ldots, q\} : R_{\geq\lambda} \text{ contains a linear order } \}$$

When $\lambda = q$, then $R_{>\lambda}$ is empty. In particular, the relation $R_{>\lambda}$ does not contain any cycles. By decreasing the value of $\lambda$, the relation $R_{>\lambda}$ will contain a cycle at some point. We are interested in the smallest value of $\lambda$ such that the corresponding strict cut-relation is acyclic.

$$\beta = \min\{\lambda \in \{0, \ldots, q\} : R_{>\lambda} \text{ is cycle-free } \}$$

The following theorems establish the links between these two relations.

**Theorem 1** *Arrow and Raynaud(1986)[1] (page 93)*
*If the constant sum property holds, then any linear order $O$ containing $R_{>\beta}$ is contained in $R_{\geq\alpha}$ and any linear order contained in $R_{\geq\alpha}$ also contains $R_{>\beta}$. Furthermore, $\alpha + \beta = q$.*

Arrow and Raynaud [1] (see axiom V' page 95) thus proposed that the compromise ranking should be a prudent order.

**Definition 1** *A prudent order $O$ is a linear order that contains $R_{>\beta}$ and is contained in $R_{\geq\alpha}$ : $R_{>\beta} \subseteq O \subseteq R_{\geq\alpha}$.*

We will denote by $\mathcal{PO} = \{O \in \mathcal{O} : R_{>\beta} \subseteq O \subseteq R_{\geq\alpha}\}$ the set of all prudent orders. The authors justify this approach to be *prudent* by the fact that ordered pairs of alternatives that belong to the relation $R_{>\beta}$ are pairs with no contradiction and a high support. If these pairs would not belong to the final compromise ranking, there would be a large and non-divided coalition against such a ranking. On the other hand, a group ranking which is not contained in $R_{\geq\alpha}$ has at least one ordered pair of alternatives $(a_i, a_j)$ such that $S_{ij}$ is strictly smaller than $\alpha$. That is why such a group ranking should be discarded.

Let us remind the reader that the transitive closure of a relation $R$, denoted by $t(R)$, is the smallest relation that is transitive and that contains $R$. Furthermore, a linear extension of a relation $R$ is a linear order $O \in \mathcal{O}$ that contains this relation: $R \subseteq O$.

**Proposition 1** *Let $O$ be a linear order. The following statements are equivalent.*

1. *$O$ is a prudent order*
2. *$O$ is an optimal solution of* $\max_{O \in \mathcal{O}} \min_{(a_i,a_j) \in O} S_{ij}$

3. *$O$ is an optimal solution of* $\min_{O \in \mathcal{O}} \max_{(a_i,a_j) \notin O} S_{ij}$

4. $t(R_{>\beta}) \subseteq O$

Point 2 states that a prudent order is a ranking that maximizes the weakest link. Equivalently, point 3 states that a prudent order is a ranking that minimizes the strongest opposition. Let us note that these results are only valid when the constant-sum property is satisfied.

One can show that the transitive closure of a cycle-free relation is a partial order (reflexive, transitive and antisymmetric relation). Since $R_{>\beta}$ is cycle-free, then, following point 4 of proposition 1 $\mathcal{PO}$ can be seen as the set of linear extensions of the partial order $t(R_{>\beta})$. This equivalence between linear extensions of a partial order and prudent orders will be useful, from a practical point of view, to perform the computations introduced in the next sections.

The properties of prudent orders have been analyzed by Debord [12] (chapter 6) and by Lansdowne [25]. Let us point out that prudent orders verify unanimity: if an alternative $a_i$ is preferred to another alternative $a_j$ in all the rankings, then $a_i$ will also be preferred to $a_j$ in all the prudent orders. Another important characteristic of prudent orders is that they are Condorcet-consistent: in case the majority relation [1] is a linear order, this will be the unique prudent order. Let us also refer the interested reader to the axiomatic characterization of the prudent order preference function presented in [24].

Prudent orders are interesting because they synthesize the profile not necessarily into one but into a whole range of possible compromise rankings. Hence they do not completely act as an aggregation operator on the profile, but rather tend to reformulate the information contained in the initial rankings.

If there are more contradictions in the initial orders, then cycles may appear more easily. This means that $\beta$ will be larger and, consequently, $\alpha$ will be smaller. As the two relations $R_{>\beta}$ and $R_{\geq\alpha}$ drift further apart, the number of prudent orders will increase, until the trivial case where every linear order on $\mathcal{A}$ is a prudent order (see for instance [26] for such an example). In fact, a direct corollary of point 4 of proposition 1 is that the size of $\mathcal{PO}$ is equal to the number of linear extensions of the partial order $t(R_{>\beta})$. This number can increase dramatically as the number of incomparable pairs in the partial order increases. In that context, let us mention Debord [12] (chapter 6), who performed simulations to estimate the number of prudent orders for small profiles. For instance, the

[1]The majority relation $M$ is a binary relation defined as follows: $(a_i, a_j) \in M \iff S_{ij} > \frac{q}{2}$.

| $S$ | $a$ | $b$ | $c$ | $d$ | min |
|---|---|---|---|---|---|
| $a$ | . | 3 | 2 | 1 | 1 |
| $b$ | 2 | . | 3 | 2 | 2 |
| $c$ | 3 | 2 | . | 3 | 2 |
| $d$ | 4 | 3 | 2 | . | **2** |

| $S$ | $a$ | $b$ | $c$ | min |
|---|---|---|---|---|
| $a$ | . | 3 | 2 | 2 |
| $b$ | 2 | . | 3 | 2 |
| $c$ | 3 | 2 | . | **2** |

| $S$ | $a$ | $b$ | min |
|---|---|---|---|
| $a$ | . | 3 | **3** |
| $b$ | 2 | . | 2 |

Figure 1: Obtaining the prudent order $dcab$ with Kohler's rule.

average number of prudent orders of profiles with 7 alternatives and 10 rankings is 134.

One way of finding a prudent order is to use Kohler's ranking rule [23], since every solution found by this rule must be a prudent order ([1], page 103). This rule can be seen as a sequential maxmin rule: at step $r$, identify the minimum along each row of $S$. Select the row such that the minimum is maximum. If there are ties, choose one arbitrarily. This row corresponds to the alternative that will be put at rank $r$ in the ranking. Delete the row and the column corresponding to the selected alternative. Furthermore, one can show ([23, 1]) that the smallest of these successive maxima corresponds to $\alpha$.

However, Kohler's rule is not sufficient to find the whole set of prudent orders (see Lansdowne [26] for a counter example). A straightforward algorithm to enumerate all the prudent orders is as follows: apply Kohler's rule once. Find the value of $\beta = q - \alpha$. Compute $R_{>\beta}$ and $t(R_{>\beta})$. Apply an algorithm that enumerates all the linear extensions of a partial order. For instance, Pruesse and Ruskey [27] proposed an algorithm that generates all the linear extensions in constant amortized time, which means in $O(|\mathcal{PO}|)$.

In figure 1, we illustrate this on the profile $u = (abcd, bcda, cdab, dabc, dcba)$. For instance, the prudent order $dcab$ will be obtained using Kohler's rule. The smallest of the coefficients selected is $\min\{2, 2, 3, 2\} = 2$. Hence, $\alpha = 2$ and $\beta = q - \alpha = 5 - 2 = 3$. The set of prudent orders thus corresponds to all the linear extensions of the $t(R_{>3}) = R_{>3} = \{(d, a)\}$. These 12 prudent orders are listed in figure 2.

Although we do not want, a priori, to discard any prudent order, it can sometimes be useful to select a few "good" prudent orders. In order to discriminate between the prudent orders, different approaches can be considered:

- Evaluate each prudent order $O \in \mathcal{PO}$ according to the sum of the outranking coefficients $\sum_{(a_i,a_j)\in O} S_{ij}$ and select the prudent orders with the largest evaluation. Let us note that, if the Kemeny order [22] is also a prudent order, this approach will select a Kemeny-optimal order. However, by adopting a sum-operator, the ordinal

| Prudent orders | $S$-coeff. | ordered $S$-coeff. | Sum |
|---|---|---|---|
| $dcba$ | (4,3,2,2,3,2) | (2,2,2,3,3,4) | 16 |
| $dcab$ | (4,3,2,3,2,3) | (2,2,3,3,3,4) | 17 |
| $dbca$ | (4,3,2,3,2,3) | (2,2,3,3,3,4) | 17 |
| $dacb$ | (4,3,2,3,2,2) | (2,2,2,3,3,4) | 16 |
| $dbac$ | (4,3,2,2,3,2) | (2,2,2,3,3,4) | 16 |
| $dabc$ | (4,3,2,3,2,3) | (2,2,3,3,3,4) | 17 |
| $cdba$ | (3,2,3,3,4,2) | (2,2,3,3,3,4) | 17 |
| **cdab** | **(3,2,3,3,4,3)** | **(2,3,3,3,3,4)** | **18** |
| $bdca$ | (2,3,2,2,4,3) | (2,2,2,3,3,4) | 16 |
| $bdac$ | (2,3,2,4,2,2) | (2,2,2,2,3,4) | 15 |
| $cbda$ | (3,2,3,2,2,4) | (2,2,2,3,3,4) | 16 |
| $bcda$ | (2,3,2,3,3,4) | (2,2,3,3,3,4) | 17 |

Figure 2: The 12 prudent orders.

aspect of the model of prudent orders will be lost.

- Among all the linear orders, prudent orders are those that maximize the weakest link. Among all the prudent orders, select the ones that maximize the second weakest link. Among, these, select the ones that maximize the third weakest link. Repeat the procedure $\frac{n(n-1)}{2}$ times or until there is only one linear order left. This procedure comes down to refining prudent orders, which are min-optimal solutions, by so-called leximin-optimal solutions (see for instance Fishburn [19] for a formal definition of the lexicographic criterion). The interest of this procedure is its complete ordinal character. It simply follows the prudent logic to its end.

This is illustrated in figure 2. In the second column of this table, a vector

$$(S_{i_1 i_2}, S_{i_1 i_3}, \ldots, S_{i_1 i_n}, S_{i_2 i_3}, \ldots, S_{i_{n-1} i_n})$$

corresponding to the prudent order $O = a_{i_1} a_{i_2} \ldots a_{i_n}$ is indicated. In the third column, this vector is ordered. One can see that the ranking $cdab$ is both optimal under a sum-operator and under the lexicographic rule.

# 3 A robust framework for using prudent orders

The basic assumption in this work is that the group agrees on the fact that the solution should be a prudent order. Although prudent orders are, in general, not unique, they will however depict a whole range of possible, potentially interesting, compromise solutions. Paradoxically, prudent orders have rarely been considered as an interesting aggregation mechanism, especially because of this multiplicity (see Debord [12], page 108 ). It will be this potential diversity of prudent orders that will be at the core of our approach.

The use in a decision aid context of ambiguous ranking rules, i.e. rules that possibly lead to more than one compromise ranking, has been addressed by Roy and Bouyssou [29] (pages 359-360) in the context of a ranking at minimal distance. First of all, the authors note that allowing multiple solutions does not really solve the problem of what ranking rule to use. Furthermore, such rules can favor ambiguities and misinterpretations, since the various ranking solutions can possibly be very contradictory. In the end, it is not a solution to our problem, since the aim is to construct one unique compromise ranking.

Although prudent orders may be appropriate in our setting, the choice of an ambiguous rule is still not an obvious task and can only be justified by a sound theoretic understanding of the ranking rule. However, in case a group actually accepts to use an ambiguous ranking rule (i.e. in our case prudent orders), then we feel that there is a need to actually exploit the existence of multiple compromise rankings and to support the group in selecting the "right" compromise ranking.

More particularly, we will suggest that the possible diversity of prudent orders will be handled by computing so-called robust conclusions, which, in the end, will support the group to select one prudent order. Following the terminology of Roy [28], a robust conclusion will be an assertion valid for all the prudent orders. This scheme is illustrated in figure 3.

Initially, the concept of robustness has been presented as a way to handle decision problems with imprecise, uncertain, ill-defined and generally badly known parameters (see Bisdorff [4] and Dias [15, 16, 17]). In our setting however, the robustness issue arises conceptually because of the non-uniqueness of a compromise ranking, which can be seen as a consequence of the difficulty and ambiguity of aggregating ordinal data.

More generally, Dias [14] distinguishes between three roles of robustness in decision aid. A first approach sees robustness as an ex-ante concern, where a robustness criterion, such as for instance the maxmin criterion, is defined and optimized. In a second approach, robustness is seen as an ex-post concern, where the various solutions that can be obtained from the various versions of the decision aid problem are analyzed, but not aggregated anymore. This is in line with the ideas of robust conclusions of Bernard Roy. Finally, in a third approach, robustness is used as a tool to progress in a decision aid process, for

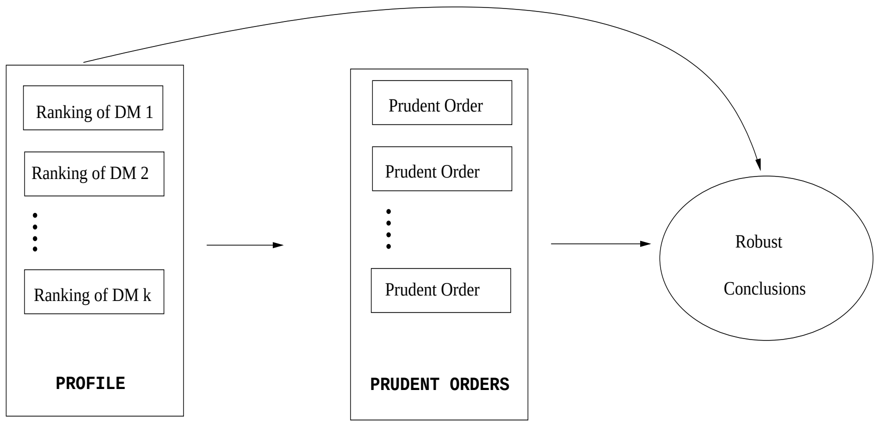



Figure 3: Computing robust conclusions.

instance by helping to refine some parameters of the decision model. This work can be seen as a contribution that combines the last two roles and uses robustness both as an ex-post exploitation procedure and as a refinement tool in a decision aid process.

Concerning the ex-post exploitation, we will compute the intersection of the prudent orders, the best and worst rank that an alternative can occupy in all the prudent orders and the maximal (or minimal) rank differences between any two alternatives.

One potential benefit of this approach relies in the fact that the information contained in the set of prudent orders will be captured while reducing the cognitive load. The solutions will be explicitly delimited, which can help the group to better understand the possibilities of comprise solutions.

Furthermore, the fact that different prudent orders can be contradictory will not be perceived as a problem anymore. The quality of the robust conclusions obtained will be inversely proportional to the degree of contradictions contained in the initial profile. Incomparabilities in the intersection, different best and worst ranks and large maximal rank differences and will point out problematic alternatives or parts of the compromise ranking. Hence, the group has to concentrate on these parts in order to reach a final compromise ranking.

Following the ideas of Dias, the robustness concept will also be used as a tool to progressively refine a decision model. In fact, by agreeing on some parts and by adapting accordingly their individual rankings, the group will gradually move toward a compromise ranking. The following four steps can thus be considered:

1. *Data collection step*
   First every decision maker proposes his individual ranking. This ranking can be obtained, for instance, by a multicriteria method of his choice.

2. *Aggregation step*
   The profile given by the group will define a set of prudent orders. If requested, within this set of rankings, an automatic procedure can select the "best" one, according to some predefined criteria, such as for instance the lexicographic criterion or the sum criterion introduced in the previous section. This ranking is then proposed to the group. Either the group accepts this solution, and the procedure stops, or the group proceeds to the next step.

3. *Analysis and discussion step*
   The set of prudent orders can be described and analyzed by various robust conclusions. Such information will help the group to understand the current possible compromise solutions.

4. *Adaptation step*
   Through the information learned from the previous step, each decision maker has the possibility to adapt his individual ranking in order to converge toward one compromise ranking. The group then proceeds back to step 1.

It is important to note that during the whole process, no solution will be imposed to the group. The robust conclusions that will be presented to the group can be considered as a guidance.

Since the number of prudent orders can increase dramatically as the number of alternatives increases and as the contradictions in the profile increases, complete enumeration will soon become infeasible, from a computational point of view. Fortunately, for all the conclusions that we will present, no complete enumeration of all the prudent orders will be necessary. This means that the robust conclusions can be directly deduced from the profile (see figure 3). In fact, the computations will be solely based on the equivalence between prudent orders and linear extensions.

Although we are mainly concerned with prudent orders, the proposed robust methodology can be extended to any other set of potentially interesting compromise rankings.

For instance, an ad-hoc solution that is used in practice is to consider all the solutions obtained by a set of different popular ranking rules (see for instance Colson [8]). In a way, a new rule is defined by combining existing rules. The idea behind this is to achieve either a confirmation of the results or to highlight at least different types of possible results since each ranking rule has its own logic and consequently treats the data from its own perspective. Such a more pragmatic approach thus also fits well into our robust scheme.

# 4 Computing robust conclusions

Instead of explicitly considering the prudent orders belonging to $\mathcal{PO}$, we are going to describe them by their intersection (section 4.1), by the possible ranks that an alternative can occupy (section 4.2) and by the maximal rank differences between two alternatives (section 4.3). Although rank ranges and rank differences can be easily understood by a group, their cardinal character should be handled with care, especially in view of the ordinal character of the prudent order model.

## 4.1 Intersection

A natural robust conclusion is to state that an alternative $a_i$ is preferred to an alternative $a_j$ in all the prudent orders of $\mathcal{PO}$:

$$(a_i, a_j) \in O \quad \forall O \in \mathcal{PO}$$

This is equivalent to stating that $(a_i, a_j)$ belongs to the intersection of all the prudent orders. We know from point 4 in proposition 1 that the set $\mathcal{PO}$ is equivalent to all the linear extensions of the partial order $P = t(R_{>\beta})$. Since the intersection of all the linear extensions of a partial order is the partial order itself (see Dushnik and Miller [18]), we have :

$$\bigcap_{O \in \mathcal{PO}} O = t(R_{>\beta}) = P$$

Hence, we do not need to explicitly enumerate all the prudent orders of $\mathcal{PO}$ in order to know their intersection. Instead it suffices to compute one transitive closure. The transitive closure can be computed by using for example Roy and Warschall's algorithm [32].

Reciprocally, given $P$, every linear extension of this partial order is a prudent order. One may show that $\forall (a_i, a_j) \in R_{>\beta}$, $S_{ij} > \frac{q}{2}$. However, it can happen that $a_i$ will be preferred to $a_j$ in all the prudent orders although $S_{ij}$ is less than $\frac{q}{2}$ (but always larger than $\alpha = q - \beta$), since $(a_i, a_j)$ has been added to $R_{>\beta}$ by transitivity.

Let us note that the $P$ relation can be interpreted as the minimal compromise that can be reached with a given profile. In case, the group members are willing to achieve a stronger compromise, they should bring closer their individual rankings with this $P$ relation. This issue will be further addressed in section 5.

## 4.2 Rank range

Let us now concentrate on the possible ranks of an alternative in the compromise ranking. By convention, if an alternative is ranked first in a linear order $O$, it has rank 1, if it is ranked second, it has rank 2 and so on. We denote by $\rho_O(a_i)$ the rank of alternative $a_i$ in the linear order $O$. Let $\rho_i^+$ and $\rho_i^-$ be the best and the worst rank a given alternative $a_i$ occupies : $\rho_i^+ = \min_{O \in \mathcal{PO}} \rho_O(a_i)$ and $\rho_i^- = \max_{O \in \mathcal{PO}} \rho_O(a_i)$. Maximal and minimal ranks in an aggregation procedure have already been suggested by Guénoche[20] in the context of determing median orders of difficult valued tournaments. However his motivation was rather algorithmic than conceptual.

Since the set $\mathcal{PO}$ consists of all the linear extensions of the partial order $t(R_{>\beta})$, one can show that (see for instance [7]) the best rank and the worst rank can be computed as follows:

$$\rho_i^+ = |N_i^+| \quad \text{where } N_i^+ = \{a_j : (a_j, a_i) \in t(R_{>\beta})\} \tag{1}$$
$$\rho_i^- = n + 1 - |N_i^-| \quad \text{where } N_i^- = \{a_j : (a_i, a_j) \in t(R_{>\beta})\} \tag{2}$$

Once $\rho_i^+$ and $\rho_i^-$ are computed, a robust conclusion would be to state that the rank of alternative $a_i$ is higher or equal than $\rho_i^+$ and smaller or equal than $\rho_i^-$. Furthermore, for each $r$ such that $\rho_i^+ \leq r \leq \rho_i^-$, we know that there exists at least one prudent order where alternative $a_i$ has rank $r$. Hence, apart from extreme rank values, the whole rank range is covered.The difference $\rho_i^- - \rho_i^+$, which is called the variability of $a_i$ by Bruggemann [7], gives an indication about the degree of contradictions or uncertainties concerning the alternative $a_i$. In fact, the bigger this difference is, the more unclear it is assigning a rank in the compromise ranking to $a_i$.

### 4.3 Maximal rank differences

We are interested in the largest possible rank difference between alternative $a_j$ and alternative $a_i$. This difference quantifies the possible advantage of an alternative $a_i$ over an alternative $a_j$.

Formally, we would like to compute:

$$\Delta_{ij}^{max} = \max_{O \in \mathcal{PO}} (\rho_O(a_j) - \rho_O(a_i))$$

More precisely, if $\Delta_{ij}^{max} \geq 0$, then this means that, at best, alternative $a_i$ is ranked $\Delta_{ij}^{max}$ positions ahead of alternative $a_j$. If $\Delta_{ij}^{max} < 0$, then this means that $a_i$ will always be at least $\Delta_{ij}^{max}$ ranks below $a_j$.

These quantities can be computed as follows:

**Proposition 2**

$$\Delta_{ij}^{max} = \begin{cases} n + 1 - (|N_j^-| + |N_i^+|) & \textit{if } (a_j, a_i) \notin t(R_{>\beta}) \\ |N_i^- \cap N_j^+| - 1 & \textit{if } (a_j, a_i) \in t(R_{>\beta}) \end{cases}$$

# 5 Adaptation step

Let us suppose that the rankings $u = (O_1, O_2, \ldots, O_q)$ provided by the experts have been aggregated into the set of prudent orders $\mathcal{PO}$. Let us note that a unique final compromise ranking has not been reached so far. The robust information introduced in the previous section should help the group to understand the current the current possibilities of compromise. We then consider the following two possibilities:

1. Several or all the group members agree to adapt their rankings. This new profile will lead to a completely new set of prudent orders. The analysis explained in the previous section can then be reapplied to that new set of prudent orders.

2. The group can agree on an ordered block partition of the alternatives. For instance, the group agrees on the fact that $a$ and $b$ occupy the first two positions of the compromise ranking, whereas the remaining alternatives have at least rank 3. This is an example of a 2-block partition, but, more generally, we can consider any ordered block partition $B_1, B_2, \ldots B_r$ of $\mathcal{A}$. Each block can then be reexamined separately.

We will now analyze the possible convergence of the set of prudent orders in this two possibilities. To do so, let us denote by $u^{new} = (O_1^{new}, O_2^{new}, \dots, O_q^{new})$ the new rankings of the group members. Let us recall that $P$ is the intersection of all the prudent orders of the initial profile. This relation $P$ will be crucial when analyzing the possible convergence of the compromise.

We say that the adaptation from $O_i$ into $O_i^{new}$ is not against the compromise if:

$$\forall a_i, a_j : \quad (a_i, a_j) \in O^{new} \wedge (a_i, a_j) \notin O \Rightarrow (a_i, a_j) \in P.$$

**Proposition 3** *If the adaptation from $O_i$ into $O_i^{new}$ is not against the compromise $\forall i = 1, \dots, q$, then $\mathcal{PO}(u^{new}) \subseteq \mathcal{PO}(u)$.*

Hence if the adaptations of all the group members are in favor of the compromise, then convergence is ensured. Hence it is reasonable to encourage the group to agree with $P$ as much as possible. However, the group should not be obliged to stick to this type of adaptations and the possibility should be given to slightly shift the focus of the set of prudent orders. Hence, it may happen that, after adapting individually, the new rankings $(O_1^{new}, O_2^{new}, \dots, O_q^{new})$ together may yield new contradictions and cycles, avoiding thus a clear convergence.

Second, let us suppose that the group applies the ordered block partition approach. Let $B_1, \dots, B_r$ be an ordered block partition of the alternatives. We say that this partition is compatible with the compromise if:

$$\forall a_i, a_j s.t. a_i \in B_k \; a_j \in B_l \; k < l \Rightarrow (a_j, a_i) \notin P$$

When examining the blocks separately, the next proposition says that the set of compromise rankings can possibly converge. Let $u_B$ be the profile restricted to the alternatives belonging to a block $B_i$. $\mathcal{PO}(u_{B_i})$ thus corresponds to the prudent orders of that restricted profile. Furthermore, let $(\mathcal{PO}(u))_{B_i}$ be the set of prudent orders of the profile $u$, but restricted to the alternatives of block $B_i$.

**Proposition 4** *Let $B_1, B_2, \dots, B_r$ be an ordered block partition compatible with the compromise. Then*

$$\forall B_i : \quad \mathcal{PO}(u_{B_i}) \subseteq (\mathcal{PO}(u))_{B_i}$$

.

# 6 Example

In order to illustrate our approach, we present an example where a group of junior researchers, mainly working in the field of information technologies in various research institutions in Luxembourg, were asked to rank a set of research domains. The problem took place in the framework of the Foresight exercise [11] organized by the Luxembourg FNR (Fonds National de la Recherche). The aim of this project is to identify socio-economic needs in order to decide on scientific research domains for Luxembourg in the medium and long term.

To achieve this goal, stakeholders were involved and, in particular, a one day workshop was organized for a group of junior researchers. During that day, 40 different research domains were put forward, amongst which 11 were finally selected as the most pertinent by the group of participants. They are listed in figure 1. The actual workshop ended at this stage.

The participants were asked after the workshop to rank these research domains. This ranking should represent their view on the prioritization of the different research domains. Let us stress that this is not an application, but only an illustration of the methodology presented in this paper. After having submitted their individual rankings, the researchers were not confronted with the results presented in this section.

Since it can be difficult for a participant of the workshop to quantify the difference of importance between two research domains, it is reasonable to use an ordinal scale. Furthermore, working with rankings as input avoids to fix a common evaluation method for the whole group.

The rankings of the 10 researchers are represented in table 2. This profile of 10 linear orders with 11 alternatives will lead to 1350 prudent orders, which means that $\frac{1350}{11!} = 0.0034\%$ of all the linear orders on 11 alternatives are prudent orders. A straightforward choice of one prudent order seems to be practically impossible. In order to go on with the decision process, several robust informations will be computed.

First of all, the intersection of all the prudent orders is represented in figure 4 (the transitivity arcs are omitted). We can learn from this figure that, for instance, $a_4$ (Electronic cooperation networks) is preferred to $a_7$ (Finance and Banking sector) in all the prudent orders.

The rank ranges are depicted in table 3. For instance the rank of $a_4$ (Electronic co-

| | |
|---|---|
| $a_1$ | Knowledge management technology |
| $a_2$ | E-Government |
| $a_3$ | IT security |
| $a_4$ | Electronic cooperation networks |
| $a_5$ | Mobile Communications |
| $a_6$ | Innovative materials and techniques in construction |
| $a_7$ | Finance and Banking sector |
| $a_8$ | Business Improvement research |
| $a_9$ | Image processing |
| $a_{10}$ | Human-Machine interface |
| $a_{11}$ | Artificial Intelligence, multi-agent systems |

Table 1: The 11 research domains

| | |
|---|---|
| uni1 | $a_4a_5a_3a_2a_8a_7a_1a_{11}a_{10}a_9a_6$ |
| uni2 | $a_3a_9a_4a_5a_2a_1a_6a_{11}a_7a_8a_{10}$ |
| uni3 | $a_4a_5a_2a_3a_1a_6a_7a_8a_{11}a_{10}a_9$ |
| uni4 | $a_1a_6a_8a_7a_{11}a_{10}a_3a_2a_5a_9a_4$ |
| uni5 | $a_{11}a_1a_9a_5a_7a_{10}a_6a_4a_3a_2a_8$ |
| iee | $a_6a_5a_7a_4a_2a_3a_8a_1a_9a_{11}a_{10}$ |
| cvce | $a_1a_{10}a_3a_4a_5a_2a_8a_7a_{11}a_9a_6$ |
| tud1 | $a_1a_4a_5a_8a_7a_3a_2a_9a_{10}a_{11}a_6$ |
| tud2 | $a_7a_8a_1a_4a_3a_9a_{10}a_{11}a_5a_6a_2$ |
| lip | $a_4a_1a_2a_5a_6a_7a_{10}a_9a_3a_{11}a_8$ |

Table 2: The rankings of the 10 researchers

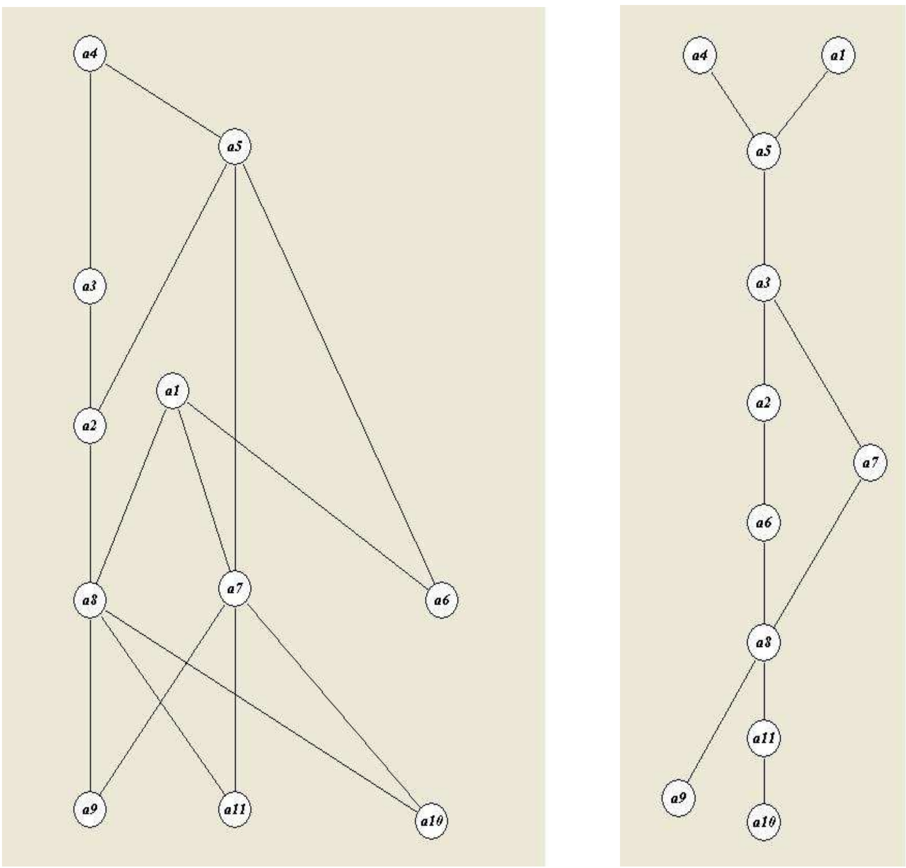



Figure 4: The intersection of all the prudent orders before (left) and after (right) the adaptation.

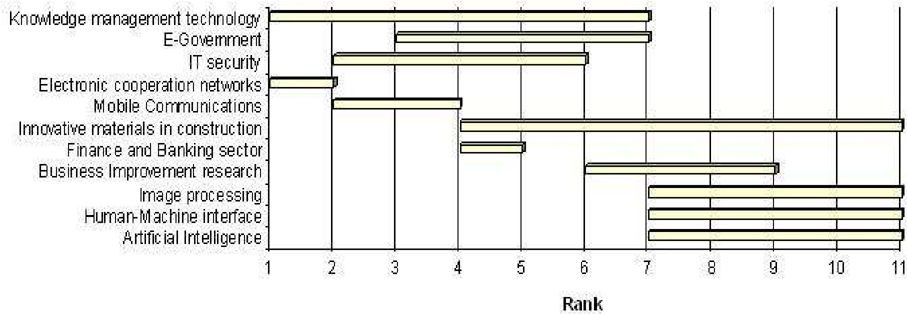



Table 3: The rank ranges

operation networks) will be either 1 or 2. On the other hand, the rank of $a_6$ (Innovative materials) seems to be unclear since, at best this alternative has rank 4 but at worst it is ranked last.

Finally, the maximal rank differences are depicted in table 4. For instance, at best, $a_1$ (Knowledge management technology) can be six ranks ahead of $a_2$ (E-Government), whereas $a_2$ can only be, at best, one rank ahead of $a_1$. There are also negative maximal rank differences. For instance, $a_{11}$(Artificial Intelligence) will always be at least 3 ranks below $a_1$.

Among the 1350 prudent orders, there are 18 orders which are optimal under the lexicographic rule. One of these rankings, could be automatically proposed to the group. Another possibility is to perform robust conclusions on these 18 rankings. For instance, the best and worst rank of each alternative in all these rankings are indicated in table 5. This information is of course consistent with the information in table 3, since lexicographic orders are a refinement of prudent orders. However, unlike for prudent orders, the extreme ranks of lexicographic optimal orders do not determine rank ranges: for instance, we know that there exists at least one lexicographic order where $a_6$ has rank 7 and at least one where $a_6$ has rank 11, but we do not have any guarantee that there also exists at least one lexicographic order where $a_6$ has rank 8, 9 or 10.

When analyzing the rank ranges of prudent orders, apart from the more problematic alternative $a_6$ (Innovative materials), let us assume that the group agrees to adopt the block partition approach in order to move toward a compromise. Let us suppose that

| | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ | $a_7$ | $a_8$ | $a_9$ | $a_{10}$ | $a_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $a_1$ | 0 | 6 | 5 | 1 | 3 | 10 | 7 | 7 | 10 | 10 | 10 |
| $a_2$ | 1 | 0 | −1 | −3 | −1 | 7 | 4 | 4 | 7 | 7 | 7 |
| $a_3$ | 3 | 5 | 0 | −1 | 2 | 9 | 6 | 6 | 9 | 9 | 9 |
| $a_4$ | 4 | 6 | 5 | 0 | 3 | 10 | 7 | 7 | 10 | 10 | 10 |
| $a_5$ | 3 | 5 | 4 | −1 | 0 | 9 | 6 | 6 | 9 | 9 | 9 |
| $a_6$ | −1 | 3 | 2 | −2 | −1 | 0 | 4 | 4 | 7 | 7 | 7 |
| $a_7$ | −1 | 3 | 2 | −2 | −1 | 7 | 0 | 4 | 7 | 7 | 7 |
| $a_8$ | −1 | −1 | −2 | −4 | −2 | 5 | 2 | 0 | 5 | 5 | 5 |
| $a_9$ | −3 | −2 | −3 | −6 | −4 | 3 | −1 | −1 | 0 | 3 | 3 |
| $a_{10}$ | −3 | −2 | −3 | −6 | −4 | 3 | −1 | −1 | 3 | 0 | 3 |
| $a_{11}$ | −3 | −2 | −3 | −6 | −4 | 3 | −1 | −1 | 3 | 3 | 0 |

Table 4: Maximal rank differences.

| alt. | $\rho^+$ | $\rho^-$ |
|---|---|---|
| $a_1$ | 1 | 2 |
| $a_2$ | 6 | 6 |
| $a_3$ | 5 | 5 |
| $a_4$ | 1 | 2 |
| $a_5$ | 3 | 3 |
| $a_6$ | 7 | 11 |
| $a_7$ | 4 | 4 |
| $a_8$ | 7 | 8 |
| $a_9$ | 8 | 11 |
| $a_{10}$ | 9 | 11 |
| $a_{11}$ | 8 | 10 |

Table 5: Best and worst rank of the lexicographic rankings.

| | $a_1, a_3, a_4, a_5$ | $a_2, a_6, a_7, a_8$ | $a_9, a_{10}a_{11}$ |
|---|---|---|---|
| uni1 | $a_4a_5a_3a_1$ | $a_2a_8a_7a_6$ | $a_{11}a_{10}a_9$ |
| uni2 | $a_3a_4a_5a_1$ | $a_2a_6a_7a_8$ | $a_9a_{11}a_{10}$ |
| uni3 | $a_4a_5a_3a_1$ | $a_2a_6a_7a_8$ | $a_{11}a_{10}a_9$ |
| uni4 | $a_1a_3a_5a_4$ | $a_6a_8a_7a_2$ | $a_{11}a_{10}a_9$ |
| uni5 | $a_1a_5a_4a_3$ | $a_7a_6a_2a_8$ | $a_{11}a_9a_{10}$ |
| iee | $a_5a_4a_3a_1$ | $a_6a_7a_2a_8$ | $a_9a_{11}a_{10}$ |
| cvce | $a_1a_3a_4a_5$ | $a_2a_8a_7a_6$ | $a_{10}a_{11}a_9$ |
| tud1 | $a_1a_4a_5a_3$ | $a_8a_7a_2a_6$ | $a_9a_{10}a_{11}$ |
| tud2 | $a_1a_4a_3a_5$ | $a_7a_8a_6a_2$ | $a_9a_{10}a_{11}$ |
| lip | $a_4a_1a_5a_3$ | $a_2a_6a_7a_8$ | $a_{10}a_9a_{11}$ |

Table 6: The adapted rankings of the 10 researchers

three blocks will be identified. A first block could contain $a_1, a_3, a_4$ and $a_5$, a second block could contain $a_2, a_7$ and $a_8$ and a third block could contain $a_9, a_{10}$ and $a_{11}$. Furthermore, since, the best rank of $a_6$ is 4, $a_6$ does not really belong to the first block. Taking a closer look at the rank differences of $a_6$, the group may agree on the fact that $a_6$ belongs also to the middle block. In fact, its ability to be much higher ranked than $a_9, a_{10}$ and $a_{11}$ does not make it a very bad alternative neither. Consequently, $a_6$ fits best in the middle part of the global ranking. Let us note that this 3 block partition is compatible with the $P$ relation.

Given the 3 block partition, the new rankings of the 10 participants can be found in table 6. We then compute the set of prudent orders separately in each block. Given this new data, the intersection of all the prudent orders can be found in figure 4 (the transitivity arcs are omitted). The final ranking is now almost complete. In fact, the still incomparable pairs are pairs that are completely indetermined in the sense that there are always five group members who prefer the first over the second alternative and five group members who prefer the second over the first alternative. In order to achieve a complete ranking, the group now has to concentrate on these pairs.

As a conclusion, the 10 group members can agree on the following compromise ranking. The most important research domains are Knowledge management technologies and Electronic cooperation networks. These two domains are then followed by Mobile Communications and IT security. Although E-Government is perceived as more important than Innovative materials, the precise rank of Finance remains unclear in the middle of the ranking. It follows Business Improvement research. Finally, Artificial Intelligence followed by Human-Machine interface research can be found at the bottom of the compromise ranking. The final position of Image processing has still to be discussed in this last part of the ranking.

## 7 Conclusion

In this paper, we suggest to better exploit the multiplicity of solutions when working with ambiguous ranking rules, and in particular with prudent orders. Instead of imposing one final compromise ranking, potentially interesting compromise rankings are described with robust conclusions, which allows the group to progressively refine the decision model and progress toward a compromise ranking in an interactive way.

In our approach, we suppose that the group has to agree on some parts of the ranking and consequently each group member has to adapt his ranking. A future research direction is to develop tools that suggests to the group "good" block. A tool may also be needed to support the group members in adapting their ranking. Another research direction is to quantify and detect the conflict (in the prudent order sense) within the group.

We presented an example with 11 alternatives and 10 group members. Although from a computational point of view, the approach seems widely applicable, a question that needs further investigation is to determine for what type of problems (number of alternatives, size of the group, degree of conflict) this prudent and robust approach actually presents the most benefits from a decision aid point of view.

Finally, a potentially useful direction of further research is to develop a decision aid tool which follows the philosophy presented in this paper, but which could handle both on the input side (i.e the ordinal data received from the group members) and on the output side (i.e. the compromise solutions) more complex preference structures than just linear orders.

## Acknowledgements

The author wants to thank Luis Dias for his valuable comments and suggestions around this work. His remarks have significantly improved the presentation of this paper.

# Appendix

Proof of proposition 1:
**Proof:**

- Equivalence between 1 and 2
  Following theorem 1, $O$ is a prudent order if and only if $O \subseteq R_{\geq\alpha}$.
  Let us show that $\alpha = \max_{O \in \mathcal{O}} \min_{(a_i,a_j) \in O} S_{ij}$. Let us suppose by contradiction that there exists $O \in \mathcal{O}$ such that $\min_{(a_i,a_j) \in O} S_{ij} > \alpha$. Consequently $O \subseteq R_{>\alpha}$. Hence there exists $\alpha' > \alpha$ such that $R_{\geq\alpha'}$ still contains at least one linear order. This contradicts the definition of $\alpha$.

  $O \subseteq R_{\geq\alpha} \iff \forall (a_i,a_j) \in O S_{ij} \geq \alpha$. This is equivalent to $\min_{(a_i,a_j) \in O} S_{ij} \geq \alpha$. This is equivalent to $O$ is optimal for $\max_{O \in \mathcal{O}} \min_{(a_i,a_j) \in O} S_{ij}$, since $\alpha = \max_{O \in \mathcal{O}} \min_{(a_i,a_j) \in O} S_{ij}$.

- Equivalence between 1 and 3
  Following theorem 1, $O$ is a prudent order if and only if $R_{>\beta} \subseteq O$.

  Let us show that $\beta = \min_{O \in \mathcal{O}} \max_{(a_i,a_j) \notin O} S_{ij}$. Let us suppose by contradiction that there exists $O \in \mathcal{O}$ such that $\max_{(a_i,a_j) \notin O} S_{ij} < \beta$. Consequently $R_{\geq\beta} \subseteq O$. Hence there exists $\beta' < \beta$ such that $R_{>\beta'}$ is acyclic. This contradicts the definition of $\beta$.

  $R_{>\beta} \subseteq O \iff \forall (a_i,a_j) \notin O\ S_{ij} \leq \beta$. This is equivalent to $\max_{(a_i,a_j) \notin O} S_{ij} \leq \beta$. This is equivalent to $O$ is optimal for $arg \min_{O \in \mathcal{O}} \max_{(a_i,a_j) \notin O} S_{ij}$ , since $\beta = \min_{O \in \mathcal{O}} \max_{(a_i,a_j) \notin O} S_{ij}$.

- Equivalence between 1 and 4
  $R_{>\beta} \subseteq O \iff t(R_{>\beta}) \subseteq t(O) = O$ (since $O$ is transitive).

□

Proof of proposition 2:
**Proof:**

- $(a_j,a_i) \notin t(R_{>\beta})$

  Since $n + 1 - (|N_j^-| + |N_i^+|) = \rho_j^- - \rho_i^+$ and since $(a_j,a_i) \notin t(R_{>\beta})$ implies that there exists at least one prudent order where $a_i$ is above $a_j$, we know that

$\Delta_{ij}^{max} \leq n + 1 - (|N_j^-| + |N_i^+|)$. We will prove that there exists one linear order $O \in \mathcal{PO}$ (i.e. one linear extension of $t(R_{>\beta})$) such that this upper bound can be reached.

A linear extension of a partial order can be constructed sequentially as follows: remove any maximal element from the partial order and rank it below the already ranked alternatives in the linear extension; stop the procedure when all the alternatives have been ranked. In our case, this procedure can be applied to the partial order $t(R_{>\beta})$. Let $N_i^+$ and $N_j^-$ be defined as in 1 and 2 and let $\bar{N}_{ij} = \mathcal{A} \backslash (N_i^+ \cup N_j^-)$. First, we will rank all the alternatives belonging to $N_i^+$ (ranking $a_i$ last ), then we will rank all the alternatives from $\bar{N}_{ij}$ and finally we will rank all the alternatives belonging to $N_j^-$( ranking $a_j$ first). One can show that at each step it is possible to find a maximal element in the relevant subset of alternatives. Let us also note that $N_i^+ \cap N_j^- = \emptyset$, because otherwise there would exist an alternative $a_t$ such that $(a_t, a_i) \in t(R_{>\beta})$ and $(a_j, a_t) \in t(R_{>\beta})$, which contradicts the fact that $(a_j, a_i) \notin t(R_{>\beta})$. Consequently, the rank of $a_i$ in this order is $|N_i^+|$, the rank of $a_j$ is $n + 1 - |N_j^-|$ and so the rank difference between $a_i$ and $a_j$ will be $n + 1 - (|N_j^-| + |N_i^+|)$.

- $(a_j, a_i) \in t(R_{>\beta})$
  Since $(a_j, a_i) \in t(R_{>\beta})$ implies that $a_j$ is always above $a_i$ in all the prudent orders of $\mathcal{PO}$. Consequently we want $a_i$ to be as close as possible to $a_j$. Since the alternatives of $N_i^- \cap N_j^+$ have to be anyway between $a_j$ and $a_i$, we must have that $\Delta_{ij}^{max} \leq |N_i^- \cap N_j^+| - 1$. We will prove that there exists one linear order $O \in \mathcal{PO}$ (i.e. one linear extension of $t(R_{>\beta})$) such that this upper bound can be reached by using the procedure described in the previous paragraph. First, we will rank the alternatives belonging to $N_j^+ \setminus N_i^-$, then we will rank the alternatives belonging to $N_i^- \cap N_j^+$ (ranking $a_i$ first and ranking $a_j$ last) and finally we will rank the remaining alternatives. □

Proof of proposition 3:

**Proof:**

Let $R_{>\lambda}$ be the cut-relation at level $\lambda$ and $\beta$ be the optimal cut-value for profile $u$. Let $R_{>\lambda}^{new}$ be the cut-relation at level $\lambda$ and $\beta^{new}$ be the optimal cut-value for profile $u^{new}$.

Since the transformation from $O_i$ into $O_i^{new}$ $\forall i$ is not against the compromise, we have that:

$$\begin{aligned} & R_{>\beta} \subseteq R_{>\beta}^{new} \subseteq R_{>\beta} \cup \{(a_i, a_j) : (a_i, a_j) \in O_k^{new} \wedge (a_i, a_j) \notin O_k\} \\ \Rightarrow\ & R_{>\beta} \subseteq R_{>\beta}^{new} \subseteq R_{>\beta} \cup \{(a_i, a_j) : (a_i, a_j) \in t(R_{>\beta})\} \\ \Rightarrow\ & R_{>\beta} \subseteq R_{>\beta}^{new} \subseteq t(R_{>\beta}) \end{aligned}$$

Since $t(R_{>\beta})$ is acyclic, $R^{new}_{>\beta}$ is acyclic, which implies that $\beta^{new} \leq \beta$. Hence :

$$R'_{>\beta'} \subseteq R'_{>\beta} \subseteq R_{>\beta}$$

This means that $\mathcal{PO}(u^{new}) \subseteq \mathcal{PO}(u)$.

Proof of proposition 4:

**Proof:** Let $O$ be a linear order on $\mathcal{A}$.Then:

$$O \in \mathcal{PO}(u) \iff R_{>\beta} \subseteq O$$

Let $O'$ be a linear order on $B$ and let $(R_{>\beta})_{B_i}$ be the cut relation of $S$ at level $\beta$ restricted to $B_i$. Since $B_i$ belongs to an ordered partition which is compatible with $P$, we can show that:

$$O' \in (\mathcal{PO}(u))_{B_i} \iff (R_{>\beta})_{B_i} \subseteq O'$$

Let $\beta'$ be the smallest value such that the strict cut relation of $S$ restricted to $B_i$ is acyclic. Since $(R_{>\beta})_{B_i}$ is acyclic, $\beta \geq \beta'$. Hence $(R_{>\beta})_{B_i} \subseteq (R_{>\beta'})_{B_i}$.

In total we have :

$$\begin{aligned} & O \in \mathcal{PO}(u_{B_i}) \\ \Rightarrow\ & (R_{>\beta'})_{B_i} \subseteq O \\ \Rightarrow\ & (R_{>\beta})_{B_i} \subseteq O \\ \Rightarrow\ & O \in (\mathcal{PO}(u))_{B_i} \end{aligned}$$

□

# References


[1] K. Arrow and H. Raynaud. *Social Choice and Multicriterion Decision-Making*. MIT Press, 1986.

[2] M.P. Beck and B.W. Lin. Some heuristics for the consensus ranking problem. *Computers and Operations Research*, 10:1–7, 1983.

[3] J.J. Bernardo. An assignment approach to choosing R&D experiments. *Decision Sciences*, 8:489–501, 1977.

[4] R. Bisdorff. Concordant outranking with multiple criteria of ordinal significance. a contribution to robust multicriteria aid for decision. *4OR*, 2:293–308, 2004.

[5] J.M. Blin. A linear assignement formulation fo the multiattribute decision problem. *Revue Française d'automatique, d'informatique et de recherche opérationelle*, 6:21–32, 1976.

[6] J.C. Borda. Mémoire sur les élections au scrutin. *Histoire de l'académie royale des sciences*, 1784.

[7] R. Bruggemann, E. Halfon, G. Welzl, K. Voigt, and C.E.W. Steinberg. Applying the concept of partially ordered sets on ranking of near shore sediments by a battery of tests. *Journal of Chemical Information and Computer Sciences*, 4:918–925, 2001.

[8] G. Colson. The OR's prize winner and the software ARGOS: how a multijudge and multicriteria ranking GDSS helps a jury to attribute a scientific award. *Computers and Operations Research*, 27:741–755, 2000.

[9] W. Cook and L. Seiford. Priority ranking and consensus formation. *Management Science*, 24:1721–1732, 1978.

[10] A.H. Copeland. A 'reasonable' social welfare function. Technical report, University of Michigan, 1951. Notes from a Seminar on Applications of Mathematics to the Social Sciences.

[11] Fonds National de la Recherche. FNR Foresight project. `http://www.fnrforesight.lu`, 2006.

[12] B. Debord. *Axiomatisation de procédures d'agrégation de préférences*. PhD thesis, Université Scientifique et Médicale de Grenoble, 1987.

[13] W. DeKeyser and J. Springael. Another way of looking at group decision making opens new perspectives. Technical Report 2002015, University of Antwerp, Faculty of Applied Economics, 2002.

[14] L. Dias. An note one the role of robustness analysis in decision-aiding processes. In *Proceedings of the 63rd Meeting of the EURO Working group on Multicriteria Decision Aid*, Porto, 2006.

[15] L. Dias and J. Clímaco. On computing ELECTRE's credibility indices under partial information. *Journal of Multi-Criteria Decision Analysis*, 8:74–92, 1999.

[16] L. Dias and J. Clímaco. ELECTRE TRI for groups with imprecise information on parameter values. *Group Decisions and Negotiation*, 9:355–377, 2000.

[17] L. Dias and J. Clímaco. Dealing with imprecise information in group multicriteria decisions: a methodology and a GDSS architecture. *European Journal of Operational Research*, 160:291–307, 2005.

[18] B. Dushnik and E. Miller. Partially ordered sets. *American Journal of Mathematics*, 63:600–610, 1941.

[19] P. Fishburn. Lexicographic orders, utilities and decision rules: A survey. *Management Science*, 20:1442–1471, 1974.

[20] A. Guénoche. Vainqueurs de Kemeny et tournois difficiles. *Mathématique, Informatique et Sciences humaines*, 133:57–65, 1996.

[21] T. Jelassi, G. Kersten, and S. Ziont. An introduction to group decision and negotiation support. In C.A. Bana e Costa, editor, *Readings in multiple criteria decision making*, pages 537–568. Springer-Verlag, 1990.

[22] J. Kemeny. Mathematics without numbers. *Daedulus*, 88:571–591, 1959.

[23] G. Kohler. *Choix multicritère et analyse algébrique de données ordinales*. PhD thesis, Université Scientifique et Médicale de Grenoble, 1978.

[24] C. Lamboray. An axiomatic characterization of the prudent order preference function. *Annales du Lamsade*, 6:229–256, 2006.

[25] Z. Lansdowne. Ordinal ranking methods for multicriterion decision making. *Naval Research Logistics*, 43:613–627, 1996.

[26] Z. Lansdowne. Outranking methods for multicriterion decision-making : Arrow's and Raynaud's conjecture. *Social Choice and Welfare*, 14:125–128, 1997.

[27] G. Pruesse and F. Ruskey. Generating linear extensions fast. *SIAM Journal of Computing*, 23:373–386, 1994.

[28] B. Roy. A missing link in operational research decision aiding: robustness analysis. *Foundations of Computing and Decision Science*, 23:141–160, 1998.

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

[30] P. Simpson. On defining areas of voter choice: Professor Tullock on stable voting. *Quarterly Journal of Economics*, 83:478–490, 1969.

[31] P. Slater. Inconsistencies in a schedule of paired comparisons. *Biometrica*, 48:303–312, 1961.

[32] S. Warschall. A theorem on boolean matrices. *Journal of the ACM*, 9:11–12, 1962.

# Duality, Robustness, and 2-stage robust LP decision models. Application to Robust PERT Scheduling

Michel Minoux*

**Résumé**

Les modèles de programmation linéaire robuste étudiés dans la littérature relèvent principalement, soit de l'incertitude par colonnes, soit de l'incertitude par lignes. Ils supposent essentiellement que les colonnes (resp : les lignes) de la matrice des contraintes ont des coefficients pouvant fluctuer, tout en restant inclus dans des ensembles d'incertitude bien définis. On commence par étudier la question de savoir si la théorie de la dualité en programmation linéaire peut être utilisée pour transformer un problème de PL robuste avec incertitude disons par lignes, en un problème de PL robuste avec incertitude par colonnes. On exhibe des exemples simples montrant clairement que le dual d'un programme linéaire robuste donné n'est en général PAS équivalent à la version robuste du dual (en considérant bien sûr que les ensembles d'incertitude pour les lignes - resp : les colonnes - du primal sont définis exactement de la même façon que les ensembles d'incertitude pour les colonnes - resp : les lignes - du dual). On poursuit l'étude de cette question de la dualité dans le contexte de la robustesse en considérant la sous-classe des programmes linéaires robustes avec incertitude sur le second membre seulement (cette sous-classe ne semble pas, jusqu'ici, avoir fait l'objet d'études significatives). Dans ce contexte nous introduisons le concept de 'modèle de programmation linéaire robuste sur deux étapes', par contraste avec le cas standard (qui pourrait être désigné sous le nom de 'modèle de robustesse sur une étape'), et nous formulons successivement (a) le dual du programme linéaire robuste ; (b) la version robuste du dual. Les expressions de la fonction objectif a optimiser dans les deux cas, bien que présentant des similitudes formelles, apparaissent clairement distinctes. De plus, du point de vue de la difficulté de résolution, un des problèmes est susceptible de résolution efficace (il s'agit d'un problème d'optimisation convexe), tandis que l'autre, en tant que problème non convexe, peut être difficile à résoudre dans le cas général. Comme application du modèle de programmation linéaire robuste à deux étapes proposé ici, nous discutons

*University Paris-6, France {Michel.Minoux}@lip6.fr

finalement le problème d'ordonnancement PERT robuste en considérant deux définitions naturelles possibles de l'ensemble d'incertitude pour les durées des tâches : le cas où l'ensemble d'incertitude est, à une mise à l'échelle près, une boule relativement à la norme Linfini ; et le cas où l'ensemble d'incertitude est, à une mise à l'échelle près, une boule de rayon maximum fixé relativement à la distance de Hamming. On montre que, dans les deux cas, le problème d'optimisation robuste peut être résolu efficacement en temps polynomial. On explique enfin pourquoi le modèle et l'approche antérieurement proposés par Bertsimas et Sim (pour le problème du plus court / plus long chemin robuste) ne pourraient s'appliquer au problème d'ordonnancement PERT robuste étudié ici, du fait de structures sous-jacentes essentiellement différentes.

**Mots-clefs :** Programmation Linéaire robuste, dualité, ordonnancement PERT robuste

**Abstract**

The various robust linear programming models investigated so far in the literature essentially appear to be based either on what is referred to as 'rowwise' uncertainty models or on 'columnwise' uncertainty models (these basically assume that the rows - resp: the columns - of the constraint matrix are subject to changes within a well specified *uncertainty set*). We first address the question of whether LP duality can be used to convert a robust LP problem with, say, rowwise uncertainty, into a robust LP with columnwise uncertainty. We provide simple examples supporting the general statement that the dual of a given robust linear programming model is NOT equivalent to the robust version of the dual (assuming of course that uncertainty for the rows - resp: the columns - of the primal, is specified in exactly the same way as uncertainty for the corresponding columns - resp: rows - of the dual). Next, we further investigate this issue of duality in the context of robustness by considering the subclass of robust LP models with uncertainty limited to the right handside only. (this subclass does not appear to have been significantly investigated so far). In this context we introduce the concept of ' two-stage robust LP model' as opposed to the standard case (which might be referred to as 'single-stage robust LP model') and we show how to derive both statements of (a) the dual to the robust model and (b) the robust version of the dual. The resulting expressions of the objective function to be optimized in both cases, though bearing some formal resemblance, appear to be clearly distinct. Moreover, from a complexity point-of-view, one appears to be efficiently solvable ( it reduces to a convex optimization problem) whereas the other, as a nonconvex optimization problem, is expected to be computationally difficult in the general case. As an application of the 2-stage robust LP model introduced here, we next investigate the *robust PERT scheduling problem*, considering two possible natural ways of specifying the uncertainty set for the task durations : the case where the uncertainty set is a scaled ball with respect to the $L_\infty$ norm; the case where the uncertainty set is a scaled Hamming ball of bounded radius (which, though leading to a quite different model, bears some resemblance to the well-known Bertsimas-Sim approach to robustness). We show that in both cases, the resulting robust optimization problem can be efficiently solved in polynomial time.

**Key words :** Robust Linear Programming, duality, robust PERT scheduling

# 1 Introduction

Various models for handling robustness objectives with respect to uncertainties on some specified coefficients in linear programming models have been proposed in the literature. We can mention Soyster (1973), Ben-Tal and Nemirovski (1998, 2000), Bertsimas and Sim (2003, 2004).

The various approaches proposed can roughly be divided into two distinct categories, depending on whether the underlying uncertainty model refers to possible fluctuations on the row vectors of the constraint matrix (we call this *'rowwise uncertainty'*), or on column vectors (we call this *'columnwise uncertainty'*).

Columnwise uncertainty was first considered by Soyster (1973). In this model each column $A_j$ of the $m \times n$ constraint matrix is either supposed to be exactly known, or is only known to belong to a given subset $K_j \subset \mathbb{R}^m$ ('uncertainty set'). The cost vector and the right handside are supposed to be certain. A robust solution is a solution which is feasible for all possible choices of the uncertain column vectors in their respective uncertainty sets. With this definition, assuming nonnegativity constraints on all variables of the LP, it can easily be shown that the problem of finding an optimal robust solution reduces to solving an ordinary LP with constraint matrix $A = (a_{i,j})$ where, $\forall i, j,$ the coefficient $a_{i,j}$ is defined as:

- $a_{i,j} = \max_{v \in K_j} \{v_i\}$ in case of a $i$th constraint of the form $\leq$
- $a_{i,j} = \min_{v \in K_j} \{v_i\}$ in case of a $i$th constraint of the form $\geq$

Note that the above maximization (or minimization) can easily be carried out if we assume the uncertainty sets either of finite cardinality (and not too big !), or closed convex.

As observed by many authors, a drawback of Soyster's model is that it usually leads to rather conservative solutions, in other words the price to pay for robustness in the above sense is often too high.

Contrasting with the above, rowwise uncertainty has attracted more interest and has been studied, among others, by Ben-Tal and Nemirovski (1998, 2000), and more recently by Bertsimas and Sim (2003, 2004).

Ben-Tal and Nemirovski start with the assumption that each row $A_i$ of the constraint matrix belongs to a known uncertainty set consisting of an ellipsoid $E_i \subset \mathbb{R}^n$ , and a solution $x \in \mathbb{R}^n, x \geq 0$ is said to be robust in this context iff it satisfies:

$$\text{for all } i : A_i x \leq b_i, \forall A_i \in E_i.$$

Ben-Tal and Nemirovski then show that finding an optimal robust solution reduces to solving a conic quadratic problem, which can be done in polynomial time. Obviously a drawback of this approach is that, due to nonlinearity of the resulting robust model, *it is not practically applicable to very large problems*.

A way to obviate nonlinearity, while retaining the idea of rowwise uncertainty, was proposed by Bertsimas and Sim (2003, 2004), considering a slightly different model of uncertainty. More precisely, they assume that each uncertain coefficient $a_{i,j}$ can take values in a given interval $[a_{i,j} - \alpha_{i,j}, a_{i,j} + \alpha_{i,j}]$ and, for each row $i$, a positive parameter $\Gamma_i > 0$ (not larger than the total number of uncertain coefficients in row $i$) is considered. A solution $x$ is then qualified as $\Gamma-$robust (in the sense of Bertsimas and Sim) iff for all $i = 1, ..., m$ this solution satisfies the $i$th constraint for all possible choices of the coefficients in row $i$ such that at most $\Gamma_i$ of the uncertain coefficients in the row are allowed to deviate from the nominal values $a_{i,j}$. (note that the above statement implicitly assumes the $\Gamma_i$ parameters to be integers, but Bertsimas and Sim show that a slightly more general definition, allowing for nonintegral values of the $\Gamma_i$'s can be handled in the same way). With this model of uncertainty, Bertsimas and Sim show that finding an optimal $\Gamma-$robust solution can be reduced to solving an ordinary linear program only moderately increased in size, thus opening the way to large scale applications. Moreover the approach readily extends to optimization problems including integrality constraints on all or part of the variables, in that case the robust version of the problem is a MIP, but, again, the resulting robust model is only moderately increased in size as compared with the original model.

## 2 Duality and columnwise/rowwise robustness

We address in this section the question of whether duality can prove in any way useful to convert a columnwise uncertain linear program into a rowwise uncertain linear program, assuming of course the same uncertainty model for the columns of the given linear program and for the corresponding rows in the dual.

Intuitively, no nice (i.e. strong) duality result is to be expected when taking into account robustness constraints since, in both the primal and the dual, there is a price to pay for uncertainty, therefore: if we maximize in the primal, the robust primal optimal solution value will be (in general strictly) less than the optimal solution value of the 'nominal' primal LP; and minimizing in the dual will lead to a robust dual optimal solution value (in general) strictly larger than the same value.

Let us illustrate the phenomenon on a small typical example. Consider the following LP (a continuous knapsack problem actually) with two uncertain coefficients $a_1$ and $a_2$ in the constraint matrix:

$$(P) \quad \begin{array}{ll} \text{Maximize} & 4x_1 + 3x_2 \\ \text{s.t:} & \\ & a_1x_1 + a_2x_2 \leq 4 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

the standard LP dual of which reads:

$$(D) \quad \begin{array}{ll} \text{Minimize} & 4u \\ \text{s.t:} & \\ & a_1u \geq 4 \\ & a_2u \geq 3 \\ & u \geq 0 \end{array}$$

Let us assume that the uncertainty set for $a_1$ is the real interval $[2, 3]$ , the uncertainty set for $a_2$ is the real interval $[1, 2]$, and let us take as definition of a robust solution in both (P) and (D) a solution which is feasible for any possible values of $a_1$ and $a_2$ in their respective uncertainty sets. Then it is easily seen that the optimal robust primal solution is $x^0 = [0, 2]$ with corresponding primal objective function value 6; and the optimal robust dual solution is $u^0 = 3$ with corresponding dual objective function value 12. This example thus clearly shows that no natural extension of the usual properties related to LP duality is to be expected in the context of robust Linear Programming. The results discussed in the following section will provide additional evidence of this.

# 3 Duality and robustness for LP's with uncertainty in the RHS

A special case of columnwise uncertainty in Linear Programming is when uncertainty only concerns the coefficients of the right handside (RHS). Such problems frequently arise in practical applications. As a typical example, we mention the robust PERT scheduling problem with uncertainty on the durations of (some of) the tasks, assuming that a robust earliest termination date has to be determined. More precisely, we want to determine the minimum total duration of the project under any possible assignment of task durations, taken in a given uncertainty set. The 2-stage robust model discussed in § 3.2 below will appear to be relevant to such applications.

Consider the following LP:

$$\begin{aligned} \max \quad & c^T x & \\ & Ax & \leq b \\ & x & \geq 0 \end{aligned}$$

and assume that the right handside b is not known exactly, but only known to belong to some uncertainty set $B \subset \mathbb{R}^m$. The set $B$ may be finite or infinite (we will introduce additional assumptions on this set when necessary).

Two distinct robustness models for LP's with uncertain RHS will be successively discussed in this section, namely single-stage robust decision models (§ 3.1) and two-stage robust decision models (§ 3.2). For both cases it will be shown that, even in the restricted situation addressed here (uncertainty in the RHS only) one cannot use standard duality theory to convert a columnwise uncertain linear program into a rowwise uncertain linear program while preserving equivalence. This will thus provide additional evidence of the fact that strong duality does not hold in general for uncertain LP's.

## 3.1 Single-stage robust (LP) decision model

We first consider the simplest case where the values of all the decision variables $x$ have to be fixed (taking into account uncertainty) before we get any kind of information on the actual realization of the uncertain parameters. In such a simple model (indeed a special case of Soyster's model) feasibility has to be ensured for any $b \in B$ , and the problem to be solved simply reduces to:

$$\begin{aligned} \max \quad & c^T x & \\ & Ax & \leq \underline{b} \\ & x & \geq 0 \end{aligned}$$

where, $\forall i, \underline{b}_i = \min_{b \in B}\{b_i\}$

The (standard) LP dual to the above problem reads:

$$\begin{aligned} \text{(D1)} \quad \min \quad & u^T \underline{b} \\ & u^T A \geq c \\ & u \geq 0 \end{aligned}$$

On the other hand, if we consider the dual to:

$$\begin{array}{lrl} \max & c^T x & \\ & Ax & \leq b \\ & x & \geq 0 \end{array}$$

we get:

$$\begin{array}{lrl} \min & u^T b & \\ & u^T A & \geq c \\ & u & \geq 0 \end{array}$$

Now consider the robust version of this dual problem where the cost vector $b$ is uncertain and can take any value in $B$. A simple and natural objective in this context is to find $u$ achieving a minimum value of $\max u^T b$ over all possible $b \in B$, thus leading to:

$$\begin{array}{lll} \text{(D2)} & \min_u \max_{b \in B} & \{u^T b\} \\ & \text{s.t:} & u^T A \geq c \\ & & u \geq 0 \end{array}$$

It is clearly realized that (D1) and (D2) are completely different optimization problems on the same solution sets since,

$$\forall u \geq 0, u^T \underline{b} \leq \max_{b \in B} \{u^T b\}$$

with strict inequality holding in the general case.

If the set B is either finite, or closed convex, it is observed that (D2) can be efficiently solved via standard convex optimization techniques since the function:

$$u \rightarrow \max_{b \in B} \{u^T b\}$$

is convex. Also, in this case, $\underline{b}$ can be efficiently computed since $\min_{b \in B}\{b_i\}$ too is a convex optimization problem, therefore problem (D1), too, can be efficiently solved.

## 3.2 Two-stage robust (LP) decision model

It frequently arises in applications that the process of decision-making under uncertainty can be decomposed in two successive steps (two-stage decision making) or more (multi-stage decision making). For simplicity of presentation, we restrict here to the case of two-stage decision making. In this case, the set of decision variables $x$ is decomposed (partitioned) into two distinct sets of variables which we denote $y$ and $z$. The $y$ variables concern the decisions to be taken in the first stage (before knowing anything about which realization of uncertainty will arise) and the $z$ variables concern the decisions to be taken in the second stage (after realization of uncertainty).

Limiting ourselves, to make the discussion easier to follow, to the case where the objective function only depends on the decision variables of the first stage, our decision problem can thus be rewritten:

$$\begin{array}{lll} \text{(I)} & \max & \gamma^T y \\ & \text{s.t:} & Fy + Gz \geq b \\ & & y \geq 0, z \geq 0 \end{array}$$

where $\gamma, b$ are vectors and $F$ and $G$ are matrices of appropriate dimensions. (Observe that the reason for restricting to the case of an objective not depending on the $z$ variables is only for the sake of simplicity in the presentation, our two-stage robust decision model would readily handle the general case of an objective depending on both the y and z variables).

Now, since the RHS $b$ in (I) is uncertain, we have to make our robustness objectives precise. In the sequel, we consider robustness for a solution y by requiring that feasibility can be ensured for any possible RHS $b \in B$ by using the second stage decision variables $z$ (by analogy with the terminology used in stochastic programming, the z variables might be referred to as 'recourse' variables). So, if we define $Y = \{y/y \geq 0$ and $\forall b \in B, \exists z \geq 0 : Gz \leq b - Fy\}$, we want to solve:

$$\max_{y \in Y} \{\gamma^T y\}.$$

Note that in the above, for any given robust solution $y$, the value taken by the $z$ variables depends on which $b \in B$ is actually realized. This is an important feature of our model which explains why it can produce less conservative solutions as compared with Soyster's model (see example given in Remark 1 below).

According to Farkas' Lemma, we know that, for fixed $b \in B$, a necessary and sufficient condition for the existence of $z \geq 0$ verifying $Gz \leq b - Fy$ is that:

$$u^T(b - Fy) \geq 0 \text{ for all } u \text{ in the polyhedral cone: } C = \{u/u^T G \geq 0 \text{ and } u \geq 0\}.$$

Denoting $u^1, u^2, ..., u^p$, the extreme rays of the above cone, the set $Y$ can equivalently be represented as the system of linear inequalities:

$$(u^j)^T Fy \leq (u^j)^T b, \forall b \in B, \forall j = 1, ...p$$

which is equivalent to:

$$(u^j)^T Fy \leq w_j = \min_{b \in B} (u^j)^T b.$$

The robust 2-stage decision problem is then reformulated as:

$$\begin{array}{lll} \text{(I)'} & \max & \gamma^T y \\ & \text{s.t:} & (u^j)^T Fy \leq w^j, \forall j = 1, ..., p \\ & & y \geq 0 \end{array}$$

Since we are interested in investigating duality in the context of robustness, let us state the (standard) dual to (I)'. So, introducing dual variables $\lambda_j, (j = 1, ...p)$, the dual to (I)' can be written as:

$$\begin{array}{ll} \min & \sum_j \lambda_j w^j = \sum_j \lambda_j \min_{b \in B} (u^j)^T b \\ \text{s.t:} & \sum_j \lambda_j F^T u^j \geq \gamma \\ & \lambda_j \geq 0, \forall j = 1, ..., p \end{array}$$

and since: $\{u / u = \sum \lambda_j u^j, \lambda_j \geq 0\} = \{u / G^T u \geq 0, u \geq 0\}$, this can be rewritten as:

$$\begin{array}{lll} \text{(DI)'} & \min_u \min_{b \in B} & u^T b \\ & \text{s.t:} & F^T u \geq \gamma \\ & & G^T u \geq 0 \\ & & u \geq 0 \end{array}$$

or , equivalently if we denote $W$ denote the set of solutions to (DI)'

$$\min_{u \in W} \min_{b \in B} \{u^T b\} \tag{1}$$

An a priori different way of using duality in our context would be to take the (standard) LP dual to (I) for fixed b, and then to carry out robustness analysis with respect to the coefficients $b$ of the objective in the dual, allowing b to take all possible values in $B$ . The LP dual to (I) reads:

$$\begin{array}{lll} \text{(DI)} & \min u^T b & \\ & \text{s.t:} & F^T u \geq \gamma \\ & & G^T u \geq 0 \\ & & u \geq 0 \end{array}$$

A natural robust version of (DI) consists in finding the dual solution $u$ minimizing the value of $u^T b$ produced by the worst possible $b$, reads:

$$\min_{u \in W} \max_{b \in B} \{u^T b\} \tag{2}$$

which is to be contrasted with (1): indeed, it is seen that the robust version of the dual significantly differs from the dual of the robust version of the initial (primal) problem (I) because the function of $u$ to be minimized is $\min_{b \in B}\{u^T b\}$ in one case, and $\max_{b \in B}\{u^T b\}$ in the other case.

It is worth observing that this structural difference between the two functions also implies a difference with respect to the practical solvability of the corresponding problems. The objective function in (2) is convex in $u$, making the robust version of (DI) efficiently solvable, whereas the objective in (1) is concave in $u$, making (DI)' and thus (by standard LP duality) (I)' too, difficult problems in the general case.

**Remark 1:** As already suggested above, the two-stage robust decision model proposed here is capable of producing less conservative solutions as compared with Soyster's model. The reason for this is that if, for a given uncertainty set B, we consider Soyster's model for problem (I), the problem to be solved is

$$\max_{y \in Y_s} \{\gamma^T y\}$$

where the set $Y_s$ is defined as $\{y/y \geq 0 \text{ and } \exists z \geq 0 : Gz \leq \underline{b} - Fy\}$ with $\underline{b}$ defined as:

$$\forall i, \underline{b}_i = \min_{b \in B}\{b_i\}$$

It is easily seen that $Y_s \subseteq Y = \{y/y \geq 0 \text{ and } \forall b \in B, \exists z \geq 0 : Gz \leq b - Fy\}$ and cases where strict inclusion holds (leading to an improved robust optimal solution value over the optimal value of Soyster's model) can easily be found, as illustrated by the following example.

In this example we consider 3 variables $y \geq 0, z_1 \geq 0$ and $z_2 \geq 0$, and 3 constraints:

$$\begin{aligned} y - z_1 \leq &\ b_1 \\ y - z_2 \leq &\ b_2 \\ z_1 + z_2 \leq &\ b_3 \end{aligned}$$

and the uncertainty set $B$ is taken as the set containing the two vectors $(1, 0, 1)^T$ and $(0, 1, 1)^T$. The objective function is to maximize $y$. It is easily checked that the set $Y_S$ corresponding to Soyster's model is in this case the real interval $[0, 1/2]$ leading to an optimal robust solution value $0.5$. On the other hand, the set $Y$ corresponding to our two-stage model is the real interval $[0, 1]$ leading to the (less conservative) optimal robust solution value 1. Indeed, the value $y = 1$ is feasible in our model because, in case $b = (1, 0, 1)^T$ occurs, we can take $z_1 = 0$ and $z_2 = 1$; and, in case $b = (0, 1, 0)^T$ occurs, we can take $z_1 = 1$ and $z_2 = 0$. (Observe, as already pointed out above, that the value taken by the $z$ variables indeed depends on which $b \in B$ is actually realized). Of course, this example does not rule out the possibility of having $Y = Y_S$ for some special instances. As will be seen in the next section, this possibility will arise in connection with the robust PERT scheduling problem, in the special case (referred to there as ***Case 1***) where the uncertainty set on the task durations is the cartesian product of a family of real intervals.

# 4 An application of the 2-stage robust LP model with RHS uncertainty: robust PERT scheduling

In this section we specialize the general two-stage robust LP decision model investigated above to robust PERT scheduling, with an uncertainty set D on the durations of the tasks, supposed to be given as a (finite or infinite) list of 'scenarios'. More precisely, we want to determine an earliest termination date which can be achieved for any realization of the task durations $d$ in a given uncertainty set $D$.

## 4.1 Formulation as a 2-stage robust LP model

Consider a PERT network represented as a directed circuitless graph $N$ in which the nodes correspond to tasks (the tasks are numbered $i = 1, 2, ..n$, the set of tasks is denoted $I$), and there is an arc $(i, j)$ with length (duration) $d_i$ whenever there is a precedence constraint stating that processing of task $j$ should not start before completion of task $i$. The set of arcs is denoted $U$. We assume that node 1 has no immediate predecessor (it thus represents the initial task) and node $n$ has no direct successor (it thus represents the terminal task of the project). Denoting $y_j (j = 1, ..n)$ the starting date for each task $j$, and assuming first that the task durations $d_i$ are exactly known, we want to minimize the total duration of the project while satisfying all precedence constraints, in other words:

$$\begin{aligned} \text{Maximize} \quad & -y_n \\ \text{s.t:} \quad & \\ & y_1 = 0 \\ & y_i - y_j \leq -d_i, \forall (i, j) \in U \end{aligned}$$

Indeed, it is easy to check that in the above, $y_1 = 0$ can be replaced by $y_1 \geq 0$, or equivalently $-y_1 \leq 0$, thus the problem can be rewritten:

$$\begin{aligned} \text{Maximize} \quad & -y_n \\ \text{s.t:} \quad & \\ & -y_1 \leq 0 \\ & y_i - y_j \leq -d_i, \forall (i,j) \in U \end{aligned}$$

This model is recognized as a special case of (I), the constraint matrix $[F, G]$ being formed by the transpose of the node-arc incidence matrix of $N$ with an additional row involving variable $y_1$ only (with associated coefficient -1). $F$ is reduced in this case to a single column (the column corresponding to node $n$ in the transpose of the incidence matrix of $N$). The right handside vector $b$ is the vector with coefficients equal to the opposite of the task durations (more specifically, the right handside coefficient for the constraint corresponding to arc $(i, j) \in U$ is equal to $-d_i$). Note that we do not state explicitly the nonnegativity conditions on $y$, since they are implied by the precedence constraints and nonnegativity of the $d_i$ coefficients.

Thus the problem is cast in a form very similar to (I) the only difference being that the nonnegativity conditions on $y$ and $z$ are dropped. The consequence of this on the analysis of § 3.2 is just that we have to consider the polyhedral cone $C' = \{u/u^T G = 0 \text{ and } u \geq 0\}$ instead of the polyhedral cone: $C = \{u/u^T G \geq 0 \text{ and } u \geq 0.\}$

Due to the special structure of the $G$ matrix arising in the PERT scheduling problem, we have the following result:

**Proposition 1:** The extreme rays of the polyhedral cone C' are in 1-1 correspondence with the characteristic vectors of the various paths between node 1 and node $n$ in $N$.

*Proof:* By observing that $G^T$ is the node-arc incidence matrix of the graph $N$ without the row associated with node $n$ but with an extra column with coefficient -1 in the first row and all other coefficients 0, it is realized that $u$ satisfying $G^T u = 0$ and $u \geq 0$ corresponds to a nonnegative flow between node 1 and node $n$ in $N$ with value equal to $u_{1,1}$, the component of $u$ corresponding to the extra column with coefficient -1 in the first row and all other coefficients 0. Therefore the extreme rays of the cone $C'$ correspond to the incidence vectors of the various paths connecting node 1 to node $n$ in $N$. ■

Let us denote $P = \{\pi^1, \pi^2, ..., \pi^K\}$ the set of all paths between $1$ and $n$ in $N$, $u^1, u^2, ..., u^K$ the corresponding characteristic vectors, the condition $(u^k)^T(b - Fy) \geq 0$ specializes to: $-\sum_{i \in \pi^k} d_i + y_n \geq 0$

(this is because, in that case: $(u^k)^T b = -\sum_{i \in \pi^k} d_i$ and $(u^k)^T Fy = -y_n$). So the condition for feasibility is that for each path $\pi^k : y_n \geq \sum_{i \in \pi^k} d_i$.

In view of this, the robust PERT scheduling problem can be reformulated as:

$$\begin{aligned} \max \quad & -y_n \\ \text{s.t:} \quad & \\ & y_n \quad \geq \sum_{i \in \pi^k} d_i, \forall \pi^k \in P, \forall d \in D \end{aligned}$$

where we recall that D denotes the uncertainty set for the task durations.

This problem therefore reduces to determining the path $\pi^k$ maximizing, over the set $P$ of all possible paths in $N$, the objective function:

$$\max_{d \in D} \{ \sum_{i \in \pi^k} d_i \}$$

in other words we want to solve:

$$\max_{\pi \in P} \max_{d \in D} \{ \sum_{i \in \pi} d_i \} \text{ (RPS)}$$

('Robust Pert Scheduling' problem)

Now, if we want to go further into the analysis of (RPS), we have to specify how the uncertainty set $D$ is defined. Of course there are many possible ways for this; we content ourselves below to examine two among the most natural possible definitions, and show that, for each of them, the above robust optimization problem (RPS) can be efficiently solved .

**Case 1:** *$D$ is a scaled ball w.r.t. the $L_\infty$ norm.*

The first easy special case is when, for each task $i$, the duration $d_i$ can take any value in a given real interval $[d_i^-, d_i^+]$ with $0 \leq d_i^- \leq d_i^+$. In this case $D$ is the cartesian product: $[d_1^-, d_1^+] \times [d_2^-, d_2^+] \times ... \times [d_n^-, d_n^+]$, which may be viewed as a scaled ball w.r.t. the $L_\infty$ norm (using component-wise scaling to have all intervals of equal width). It is easily seen that an optimal robust solution for problem (RPS) can be obtained in this case by looking at a longest path (critical path) in $N$ when each of the tasks $i$ is assigned the longest possible duration $d_i^+$.

As an illustration of the above, consider the following example with $n = 7$ tasks, where the graph of precedence constraints has the following arcs: $(1, 2)$, $(1, 3)$, $(2, 3)$, $(2, 5)$, $(2, 6)$, $(3, 4)$, $(3, 7)$, $(4, 5)$, $(5, 7)$ and $(6, 7)$. Thus task 2 cannot be started before completion of task 1, etc. Also note that the tasks are numbered according to a topological ordering of the graph, since there is no arc $(i, j)$ with $i > j$. The associated intervals $[d_i^-, d_i^+]$ for the durations of the tasks are the following:

| Task 1 | Task 2 | Task 3 | Task 4 | Task 5 | Task 6 |
|---|---|---|---|---|---|
| $[2, 4]$ | $[4, 8]$ | $[3, 6]$ | $[4, 8]$ | $[4, 8]$ | $[8, 16]$ |

Task 7 is not shown in the above table because it is a dummy task (of duration 0, without uncertainty) representing the end of the schedule. It is easy to see that in this example, the optimal solution to the (RPS) problem has duration 34 and corresponds to the critical path $(1, 2, 3, 4, 5, 7)$. Indeed 34 is the earliest achievable termination date if we require that the schedule remains feasible for any possible choice of the task durations in the cartesian product $[d_1^-, d_1^+] \times [d_2^-, d_2^+] \times ... \times [d_6^-, d_6^+]$. This corresponds to the situation where the duration of each task $i$ is $d_i^+$.

**Case 2:** *$D$ is a scaled Hamming ball of bounded radius* $\Gamma$.

Here again we assume that, for each task $i$, the duration $d_i$ can take any value in a given real interval $[d_i, d_i + \Delta_i]$ with $d_i \geq 0$. $d_i$ is called the nominal value of the duration for task $i$, $d_i + \Delta_i$ being the possible extreme (or worst-case) value for the task duration. As is actually the case in many practical applications, it is unlikely that all tasks simultaneously take on worst-case values. To take this observation into account, we will impose an upper bound $\Gamma$ on the number of task durations which are allowed to take on a worst-case value, given that all task durations which do not take on a worst-case value are assumed to be equal to their nominal value. More formally, associating with each task $i$ a 0-1 integer variable $u_i$ , the uncertainty set $D$ corresponding to this definition is:

$$D = \{\theta = (\theta)_{i=1,...,n} / \theta_i = d_i + \Delta_i u_i (i = 1, ..., n) \text{ such that: } \sum_{i=1}^{n} u_i \leq \Gamma, u_i \in \{0, 1\}, \forall i\}$$

As can be seen from the above definition, in the special case where all $\Delta_i$ are equal to 1, $D$ is recognized as the Hamming ball of radius $\Gamma$ centered at $d = (d_i), i = 1, ..n$ (in other words, $\theta - d$ can be any 0-1 vector with at most $\Gamma$ components equal to 1). When the $\Delta_i$'s take on arbitrary positive values, the Hamming ball structure is still present after applying scaling to each component $i$ with respect to the corresponding $\Delta_i$ value.

We note here that, in spite of the fact that the definition above is close in spirit to the concept of uncertainty suggested by Bertsimas and Sim (2003, 2004), our model is fairly different from the one studied by these authors, since they restrict themselves to rowwise uncertainty, whereas in our robust PERT scheduling problem, we have uncertainty on the RHS only (a special case of columnwise uncertainty). For more detailed discussion of this issue, see §4.2 below.

We now show that, with the above definition of the uncertainty set $D$, problem (RPS) can be efficiently solved via a dynamic programming recursion. To that aim, we will consider the problem with parameter $\Gamma$ as only one representative of the class of problems (RPS$[i, k]$) for $i$ running from 1 to $n$ (the number of tasks) and $k$ running from 0 to $\Gamma$. More precisely, assuming that the tasks are numbered according to a topological ordering of the circuitless graph $N$ , (RPS$[i, k]$) consists of the robust PERT scheduling subproblem corresponding to the subset of tasks 1, 2,...,$i$, the durations of at most $k$ of which are allowed to take on their worst-case values. The case k=0 (no deviation allowed) corresponds to the usual PERT scheduling problem in terms of the nominal values for the task durations. We denote $v^*[i,k]$ the optimal objective function value for problem (RPS$[i,k]$), and for any task $i$, we denote $Pred[i]$ the set of tasks $j$ such that $(j,i)$ is an arc of $N$ (the set of direct predecessors of node $i$). Bellman's optimality principle then leads to the following dynamic programming recursion:

$$\forall i \in [1,n], \forall k = 0,1,...\Gamma : v^*[i,k] = \max_{j \in Pred[i]} \max\{v^*[j,k] + d_j ; v^*[j,k-1] + d_j + \Delta_j\} \tag{3}$$

The optimal value of the robust PERT scheduling problem we are interested in is then: $v^*[n,\Gamma]$. The rationale behind Eq. (3) can easily be explained as follows. Consider the set of all paths from 1 to $j \in Pred[i]$. The duration of arc $(j,i)$ has nominal value $d_j$ and worst-case value $d_j + \Delta_j$. The maximum duration of a path from 1 to $i$ through $j$ with at most $k$ tasks allowed to deviate from their nominal values can be obtained: either by allowing for at most $k$ deviations on the subset of tasks $\{1,2,..j\}$ and taking the nominal duration for arc $(j,i)$; or by allowing at most $k-1$ deviations on the subset of tasks $\{1,2,..j\}$ and taking the worst-case duration for arc $(j,i)$. Thus the optimal value for node $i$ via node $j$ is the maximum value among these two alternatives, and the optimal value for node $i$ is the maximum taken on the set of all direct predecessors of $i$. Obviously, solving the recursion (3) is achieved in polynomial time $O(m \times n)$, where $m$ is the number of arcs and n the number of nodes of the PERT network; more precisely the complexity is $O(m \times \Gamma)$ where $\Gamma$, the parameter defining the uncertainty set, is at most $n$, the number of tasks (but often significantly smaller than $n$ in practical applications).

Let us illustrate the above on the same 7 task example as the one considered to illustrate case 1. We thus consider the same intervals for the task durations, the lower bound of each interval representing the nominal task duration, and the upper bound representing the worst-case duration. For $\Gamma = 3$, application of the recursion (3) leads to the $v^*[i,k]$ values shown in the following table.

| | $i=2$ | $i=3$ | $i=4$ | $i=5$ | $i=6$ | $i=7$ |
|---|---|---|---|---|---|---|
| $k=0$ | 2 | 6 | 9 | 13 | 6 | 17 |
| $k=1$ | 4 | 10 | 13 | 17 | 10 | 22 |
| $k=2$ | 4 | 12 | 16 | 21 | 12 | 26 |
| $k=3$ | 4 | 12 | 18 | 24 | 12 | 29 |

For instance the value $v^*[4,2] = 16$ corresponds to the path $(1,2,3,4)$ with 3 arcs of nominal durations 2, 4 and 3 respectively. If, in this path, two arcs out of three are allowed to take on their worst-case durations, the worst case (Max) is obtained when task 2 has duration 8 and task 3 has duration 6 (task 1 keeping its nominal duration 2), the resulting length of the path being: 8 + 6 + 2 = 16. Let us also illustrate how the recursion (3) works for computing e.g. $v^*[7,2]$ and $v^*[7,3]$. We have:

$$\begin{aligned} v^*[7,2] &= \max\{v^*[3,2]+3; v^*[3,1]+6; v^*[5,2]+4; v^*[5,1]+8; \\ &\qquad v^*[6,2]+8; v^*[6,1]+16\} \\ &= \max\{15,16,25,25,20,26\} = 26. \end{aligned}$$

The maximum above is obtained for $j = 6$ and the corresponding optimal path is $(1,2,6,7)$.

Similarly we have:

$$\begin{aligned} v^*[7,3] &= \max\{v^*[3,3]+3; v^*[3,2]+6; v^*[5,3]+4; v^*[5,2]+8; \\ &\qquad v^*[6,3]+8; v^*[6,2]+16\} \\ &= \max\{15,18,28,29,20,28\} = 29. \end{aligned}$$

The maximum above is obtained for $j = 5$ and the corresponding optimal path is $(1,2,3,4,5,7)$. It is thus seen that, depending on the choice of the control parameter $\Gamma$, various optimal paths are obtained which, of course, may differ from the optimal solution to the non-robust PERT scheduling problem (considering only the nominal values for the task durations). Also, observe that the value $v^*[7,3] = 29$ corresponds to a less conservative robust situation as compared with the one obtained in case 1 above.

## 4.2 Differences with Bertsimas and Sim's approach

We now turn to show that, in spite of the similarity in the definition of the uncertainty sets, the robust version of the PERT scheduling problem investigated here is essentially different from the model proposed by Bertsimas and Sim [3] for the robust version of the shortest path problem. From an abstract point-of-view, the difference basically stems from the fact that, in our case, we are faced with a LP problem with uncertainty on the right handside, whereas Bertsimas and Sim address a LP problem with uncertainty on the cost coefficients. However, to further understand the source of this difference, we show below which difficulties would arise if we wanted to apply the Bertsimas-Sim approach to the robust longest (critical) path problem on a directed circuitless graph $G$.

Following these authors, the robust shortest s-t path problem in G with uncertainty parameter $\Gamma$ (assuming $\Gamma \in \mathbb{N}$) is formulated as:

$$\text{(RSP)} \quad \min_{x \in X}\{\sum_{(i,j)\in U} c_{i,j}x_{i,j} + \max_{S\subseteq U, |S|\leq\Gamma} \sum_{(i,j)\in S} \Delta_{i,j}x_{i,j}\}$$

where $X$ denotes the set of incidence vectors of all s-t paths in $G$; $c_{i,j}$ denotes the nominal cost of arc $(i, j)$ and $c_{i,j} + \Delta_{i,j}$ is the worst-case cost of arc $(i, j)$. After transformation of (RSP) using the duality theorem to convert the second term in the brackets into a minimization, the problem is reformulated as a standard LP , the solution of which reduces to $m + 1$ aplications of a standard shortest path algorithm. Observe that one of the reasons for all the above to work so nicely is that the second term in the brackets, as a function of $x$, is convex in $x$ , since it is the pointwise maximum of a finite number of linear functions.

The above approach is still valid if, instead of looking for an optimum robust minimum cost path, we were looking for an optimum robust maximum benefit path: $c_{i,j} > 0$ being interpreted as a reward associated with the use of arc $(i, j)$, the effect of uncertainty being to reduce the nominal reward $c_{i,j}$ by the amount $\Delta_{i,j}$. The problem would then take the form:

$$\max_{x \in X} \{ \sum_{(i,j) \in U} c_{i,j} x_{i,j} - \max_{S \subseteq U, |S| \leq \Gamma} \sum_{(i,j) \in S} \Delta_{i,j} x_{i,j} \}$$

which is essentially analogous to the above robust minimum cost path, up to a change in the signs of the coefficients in the objective (still assuming, of course, that the graph $G$ under consideration is circuitless). In particular, we note that the function to be maximized is concave, so we still have a convex optimization problem.

By contrast, the robust PERT scheduling problem addressed in the present paper is formulated as:

$$\max_{x \in X} \{ \sum_{(i,j) \in U} c_{i,j} x_{i,j} + \max_{S \subseteq U, |S| \leq \Gamma} \sum_{(i,j) \in S} \Delta_{i,j} x_{i,j} \}$$

(with $c_{i,j} > 0$ and $\Delta_{i,j} > 0$).

It is then readily observed that this problem consists in maximizing a convex function of $x$ on $\{0, 1\}^m$, and it is well-known that this cannot be simply reduced to ordinary linear programming as is the case for Bertsimas and Sim's approach. Thus robust PERT scheduling may be viewed a typical illustration of the big differences between models featuring rowwise uncertainty and models featuring columnwise uncertainty in robust Linear Programming.

# 5 Conclusions

In this paper, various Robust Linear Programming problems have been investigated, and the question of whether LP duality can still be used to help in solving such problems has been addressed and answered negatively. Among the problems considered, Robust Linear Programming problems with uncertainty in the right handsides only, have been recognized as an interesting sub-class of problems, for which the solution techniques should not confine themselves to the classical approach proposed by Soyster (1979). In this respect we have been lead to propose a new class of robust LP models referred to here as ’Two-Stage Robust Decision Models’ which can be expected to lead to less ’conservative’ optimal robust solutions than those usually obtained from Soyster’s model. In order to show the practical usefulness of this 2-Stage model, a specialization to robust PERT scheduling has been discussed, leading , under two natural ways of defining the uncertainty set w.r.t. the task durations, to efficient solution methods. Also some fundamental difference between our approach to robust PERT scheduling and the one proposed by Bertsimas and Sim [3] in the context of the robust shortest path problem has been pointed out. We think that many other possible applications of this 2-Stage robust modeling approach would deserve further investigations, for instance in dynamic inventory management, optimal resource allocation problems, telecommunication problems, etc. This will be the subject of future research.

## Acknowledgments

We thank the referees for their careful reading and constructive comments.

## References

[1] BEN-TAL A., NEMIROVSKI A. (1998) Robust Convex Optimization, Math. Oper. Res., 23, pp. 769-805.

[2] BEN-TAL A., NEMIROVSKI A. (2000) Robust Solution of Linear Programming Problems Contaminated with Uncertain Data, Math. Prog., 88, pp. 411-424.

[3] BERTSIMAS D., SIM M. (2003) Robust Discrete Optimization and Network Flows, Math. Prog. B, 98, pp. 49-71.

[4] BERTSIMAS D., SIM M. (2004) The Price of Robustness, Oper. Res. 52, 1, pp. 35-53.

[5] KOUVELIS P., YU G. (1997) Robust Discrete Optimization and its Applications, Kluwer Acad. Publ., Boston.

[6] MULVEY J. M., VANDERBEI R. J., ZENIOS S.A. (1995) Robust Optimization of Large Scale Systems, Oper. Res. 43,2, pp. 264-281.

[7] SOYSTER A.L.(1973) Convex Programming with Set-Inclusive Constraints and Applications to Inexact Linear Programming, Oper. Res. 21, pp. 1154-1157.

[8] SOYSTER A. L. (1979) Inexact Linear Programming with Generalized Resource Sets, EJOR, 3, pp. 316-321.

# A robust 2-stage version for the STEINER TREE problem

Vangelis Th. Paschos[*], Orestis A. Telelis[†], Vassilis Zissimopoulos[†]

**Résumé**

Dans cet article, nous considérons une version robuste pour l'arbre de Steiner. Sous cette version, le problème est défini dans un cadre en deux étapes sur un graphe complet à arêtes pondérées dont les sommets sont associés avec des probabilités de présence en seconde étape. Une solution réalisable pour la première étape sur le graphe d'entrée peut devenir non-réalisable pour la seconde étape, quand quelques sommets disparaissent. Dans ce cas, une « stratégie de modification » est conçue qui transforme une solution partielle en une solution réalisable pour la seconde étape. L'objectif est de concevoir un algorithme qui calcul une solution pour la première étape (cette solution s'appelle solution *a priori* ou « d'anticipation ») de qui minimise la fonction objectif du problème robuste. Une caractéristique importante de ce modèle robuste est que la stratégie de modification est partie du problème. Nous recherchons de résultats de complexité et d'approximation en utilisant une stratégie de modification basée sur l'algorithme du parcours en profondeur.

**Mots-clefs :** Approximation, Arbre de Steiner, Complexité, Graphe, Optimisation probabiliste, Robustesse

**Abstract**

In this paper we consider a robust version for STEINER TREE. Under it, the problem is defined in a two-stage setting over a complete weighted graph whose vertices are associated with a probability of presence in the second stage. A first-stage feasible solution on the input graph might become infeasible in the second stage, when

[*]LAMSADE, CNRS UMR 7024 and Université Paris-Dauphine. `paschos@lamsade.dauphine.fr`. Work partially performed while the author was with the Division of Theoretical Computer Science of the Department of Informatics and Telecommunications at the University of Athens

[†]Division of Theoretical Computer Science, Department of Informatics and Telecommunications, University of Athens. `{telelis,vassilis}@di.uoa.gr`

certain vertices of the graph fail (with the specified probability). Therefore, a well defined *modification strategy* is devised which transforms a partial solution to a feasible second-stage solution. The objective is to devise an algorithm for the first-stage solution (sometimes called the *a priori* or *anticipatory* solution) so that the expected second-stage solution cost is minimized. An important feature of this framework is that the modification strategy is essentially a part of the problem, while algorithmic treatment is required in the construction of the anticipatory solution. We provide complexity and approximation results regarding a modification strategy based upon the well known depth-first-search algorithm.

**Key words :** Approximation, Complexity, Graph, Probabilistic optimization, Robustness, Steiner tree

# 1 Introduction and motivation

Given an edge-weighted complete graph $G(V, E, \vec{w})$, where $\vec{w}$ is a metric $|E|$-vector that represents the edge weights, and a subset of "terminal" vertices $T \subseteq V$, the STEINER TREE problem consists of determining a minimum total-cost tree $S$ connecting the vertices of $T$. We study here the following 2-stage optimization stochastic model. Consider an instance $G(V, E, \vec{w})$ of STEINER TREE and assume that our problem is not to be necessarily solved on the whole $G$, but rather on a (unknown a priori) subgraph $G'$. Suppose that any vertex $v_i \in V$ has a probability $p_i$, indicating how vertex $v_i$ is likely to be present in the final subgraph $G'$. Suppose also that any $v_i \in T$ has presence-probability 1. Consider finally that once $G'$ is specified, the solver has no opportunity to solve it directly (for example assume that she/he has to react quasi-immediately, so no sufficient time is given her/him). Then, in order to tackle such constraints, the solver computes an *anticipatory Steiner tree* $S$ in $G$, i.e., a solution for the entire instance $G$, and once $G'$ becomes known, she/he modifies $S$ by means of an efficient algorithm, in order to get a tree $S'$ that remains a Steiner tree with respect to $G'$. The objective in such approach is to determine an initial Steiner tree $S$ for $G$ such that, for any subgraph $G'$, induced by a subset $V' \subseteq V$, that will be presented for optimization, the obtained Steiner tree $S'$ respects some pre-defined quality criterion (for example, it is optimal for $G'$, or it achieves, say, approximation ratio of a certain level).

Acquisition, validation and consistency of input data are tackled in almost any operations research application. Although several well established theoretical models exist for problems arising in real world, direct application of these models may be difficult or even impossible due to incompleteness of data, or due to their questionable validity. Occasionally, one may be asked to produce an optimal operational design even before a complete deterministic picture of input data is provided, but only based on estimations and statistical measures. There are several applications where it might be impossible to obtain a current snapshot of the required information, since this information may be subject to constant high-rate change.

Several optimization frameworks have been introduced by the operations research community for handling these deficiencies, the most known being the *stochastic programming* (see [6, 8, 27] for basics, [28] for latest news, bibliography, and related software and [13, 14, 16, 29, 31, 32, 33] for recent hardness results and approximation algorithms) and the *robust discrete optimization* (see [3, 9, 19, 21, 22] for details). These frameworks constitute a means of structuring uncertainty and taking its existence into account during the optimization process. Robustness of the designed solution from both feasibility and cost perspectives in the presence of uncertainty is the main purpose of devising these frameworks during an operational design process.

The model we deal with in this paper is a 2-stage optimization stochastic model. For graph-problems it can be sketched as follows. Consider an instance $G(V, E)$ of an optimization graph-problem $\Pi$ and assume that instead of $G$ $\Pi$ will need to be solved on an unknown a priori subgraph $G'$ of $G$. Any vertex $v_i \in V$ has a presence probability $p_i$ (vertex-probabilities are independent). Assume also that a more or less efficient algorithm $\mathtt{M}$ is given, modifying any solution $S$ for $\Pi$ on $G$ into a solution $S'$ for $\Pi$ in any subgraph $G'$ of $G$ (this algorithm is usually called *modification strategy*). The objective is to determine an anticipatory solution $S^*$ for $G$ such that, for any subgraph $G'$, induced by a subset $V' \subseteq V$, that will be presented for optimization, the solution $S'$ respects some pre-defined quality criterion. Such a goal amounts to compute an anticipatory solution $S$ optimizing the expectation of its adaptation to any $V' \subseteq V$, i.e., optimizing the sum of the occurrence probability of $V'$ multiplied by the value of $S'$ over any $V' \subseteq V$. Indeed, under the hypotheses just stated, the objective is to determine some solution $S^*$ optimizing the quantity (called also *functional*):

$$E(G, S, \mathtt{M}) = \sum_{V' \subseteq V} \Pr\left[V'\right] m\left(G\left[V'\right], S'\right) \tag{1}$$

where $m(G[V'], S')$ denotes the cost of $S'$ in $G[V']$[1] and $\Pr[V']$ the occurrence probability of $V'$ as second-stage portion expressed by: $\Pr[V'] = \prod_{v_i \in V'} p_i \prod_{v_i \in V \setminus V'}(1 - p_i)$. The optimization goal for $E(G, S, \mathtt{M})$ is identical to the one for the (deterministic) original problem $\Pi$. The derived stochastic problem will be denoted by PROBABILISTIC $\Pi$. It can easily be seen by the discussion just above that for an underlying deterministic problem $\Pi$, different modification strategies give rise to different probabilistic models of $\Pi$. In other words, for a problem $\Pi$ any modification strategy induces its own probabilistic counterpart of $\Pi$[2].

[1]Quantity $m(G[V'], S')$ depends indeed also on $S$ since it is the fitting of $S$ in $G[V']$; in this sense, $m(G[V'], S')$ should be rather written as $m(G[V'], S', S)$; but, for simplicity, this dependance will be omitted.

[2]In this sense, the correct way to denote a robust version of a combinatorial optimization problem $\Pi$

This model is originally introduced in [17, 4] under the name *a priori probabilistic combinatorial optimization* as a means of structuring and handling uncertainty in the validity of input data and of tackling robustness requirements for the solutions computed on such data. Since then, several well-known combinatorial optimization problems have been treated in this framework, such as the *longest path* ([23]), the *maximum independent set* ([24]), and the *minimum coloring* ([25, 7]). Less recent studies include the *shortest path* ([18]), the *minimum-cost spanning tree* ([5]) and the *travelling salesman* ([17]). Several examples of natural situations concerning planning, timetabling, etc., that can be modelled as probabilistic combinatorial optimization problems are given in [25, 26].

Let us note that a complementary framework to the one of the a priori optimization, is the *reoptimization* consisting of solving ex nihilo and optimally the portion of the instance presented for optimization. Reoptimization is introduced in [17]. The functional for such a process can be expressed as:

$$E^*(G) = \sum_{V' \subseteq V} \Pr\left[V'\right] \operatorname{opt}\left(G'\right) \tag{2}$$

where $\operatorname{opt}(G')$ is the value of an optimal solution for $\Pi$ in $G'$ and, as previously, $G' = G[V']$. Obviously, for any modification strategy $\mathtt{M}$, denoting by $S^*$ the optimal anticipatory solution of PROBABILISTIC $\Pi$ (under $\mathtt{M}$) and assuming that $\Pi$ is a minimization problem:

$$E^*(G) \leqslant E\left(G, S^*, \mathtt{M}\right) \tag{3}$$

Study of PROBABILISTIC STEINER TREE can be motivated by the following real-world application arising in the design of wireless networks. These networks consist of a set of nodes each transmitting a signal using a certain power level, thus being able to communicate with another network node within a range determined by the level of used power. Therefore the paradigm of multi-hop communication is inherent in such networks. That is if a node $u$ wishes to send a message to some other node $v$, then this message is forwarded progressively from $u$ to $v$ by intermediate nodes.

Wireless networks lack stable infrastructure, therefore it has been proposed in the literature ([35]), that some nodes of the network should collaborate to simulate the functionality of such an infrastructure by means of a *network backbone* (also called spine). A backbone is a set of interconnected network nodes which handles routing of transmitted messages, monitoring of the network's state, acquisition and cross-processing of sensed data (in the case of sensor networks) and several other application-specific labors. Each node of the network is assigned to a backbone node. Backbone nodes are interconnected efficiently via a tree structure emerging as a result of appropriate tuning of their power levels of transmission (see, for example, [20] for some models concerning optimization of

(under the settings described) is by PROBABILISTIC $\Pi(\mathtt{M})$; however, since throughout the paper the modification strategy used is fixed, we simplify things by just using PROBABILISTIC $\Pi$ instead.

transmission power). Each node wishing to send a message to another node simply sends this message to the backbone node it is assigned to. Subsequently the message is routed through the backbone to the receiver.

Efficient connectivity of the backbone nodes can be achieved by a solution to the STEINER TREE problem ([20]). However, nodes of a wireless network are often subject to failures caused by environmental factors, energy depletion etc. Furthermore, it is a common monitoring process (performed by the backbone nodes) to measure the probability of a wireless network node being available. Therefore the PROBABILISTIC STEINER TREE problem seems in this case a more appropriate model as a means of providing connectivity to the backbone nodes.

STEINER TREE is well known to be **NP**-hard. The first approximation algorithm for it appeared in [1] (see also [12, 34]) and consists first of a shortest path computation for all pairs of terminal vertices and next of a computation of a minimum-cost spanning tree over the shortest-path weighted complete graph with vertex set $T$. The approximation ratio achieved by this algorithm is bounded above by 2.

This result has been improved in [30] for metric complete graphs, down to 1.55 in complete metric graphs and to 1.28 for complete graphs with edge costs 1 and 2. Recently, a robust optimization model, quite different from the model we tackle here, called *demand-robust Steiner tree* has been introduced and studied in [10].

Finally, let us note that STEINER TREE has also been studied extensively under the paradigm of stochastic programming; for the best known approximation algorithms in this framework see, e.g., [11, 15].

In what follows, we mainly study the complexity and approximation of a robust model for STEINER TREE derived by a modification strategy based upon the depth-first-search algorithm ([2]). Given an anticipatory solution $S$ for PROBABILISTIC STEINER TREE(M) its approximation ratio is defined by $E(G, S, \mathtt{M})/E^*(G)$, where $E(G, S, \mathtt{M})$ and $E^*(G)$ are defined by (1) and (2), respectively.

In Section 2 we formally introduce the model tackled, we study its objective function (also called *functional*), the complexity of its computation (let us note that since it carries over all the possible subsets of the input vertex-set, the functional entails an exponential number of additive terms in such a way that the complexity of its numerical computation is not immediately polynomial) and we discuss approximation issues for this model.

# 2 Robustness for STEINER TREE under the depth-first-search modification strategy

## 2.1 The derived problem and its complexity

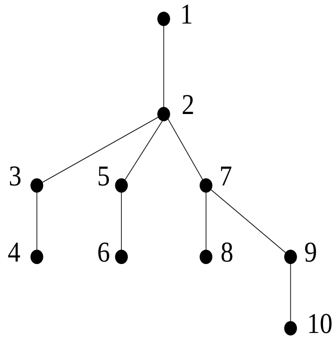
We use the following modification strategy, called DFS. Given an anticipatory Steiner tree $S \subseteq E$ (represented by the set of its edges), DFS first orders the edges of $S$ by engaging a depth-first search (DFS) starting from an arbitrary leaf-vertex of $S$. Each vertex of $S$ is assigned a number equal to the order of its visitation by the DFS. The tree $S$ is thus partitioned into maximal edge-disjoint paths, whose edges are ordered by the numbering of their end-vertices. Let $\mathcal{P}(S) = \{P_1, P_2, \ldots, P_k\}$ be the set of these paths, where the paths are ordered in the order their edges appear in $S$. We represent $S$ as an ordered multi-set of vertices $\mathcal{L}$ by placing in $\mathcal{L}$ the endpoints of the edges of $P_i$, $i = 1, \ldots, k$, following the order they appear in $P_i$ and the ordering of $\mathcal{P}(S)$. It is clear that some vertices will appear more than once in $\mathcal{L}$. When a vertex subset $V' \subseteq V$ realizes in second-stage, DFS discards from $\mathcal{L}$ all copies of vertices not present in $V'$. This causes $S$ to break down in components. Let $S''$ be the surviving edge subset of $S$ in $G[V'] = G'(V', E')$. Then, DFS traces the surviving portion $\mathcal{L}'$ of $\mathcal{L}$ and, for any two consecutive vertices $v_i$ and $v_j$ in $\mathcal{L}'$, if $v_i$ is not connected to $v_j$ and $i < j$, then edge $(v_i, v_j)$ is inserted in $S''$. This process causes reformation of all edge-disjoint paths in $\mathcal{P}(S)$ and results into a feasible Steiner tree $S'$ for $G'$. It can be immediately seen that the complexity of DFS is linear with $n$ (the order of the input graph).

Let us note that the hypothesis that the input graph $G$ is complete is a sine qua non condition for the existence of feasible solutions in any of its induced subgraphs. Indeed, if $G$ is not complete there may exist subsets $V' \subseteq V$ inducing non connected subgraphs in which patching of the components of $S''$ is impossible.

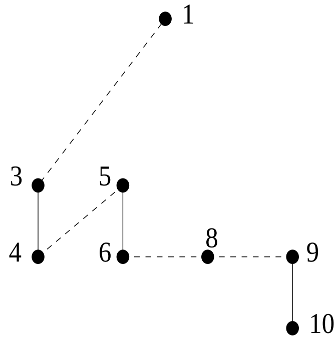
**Example 1.** Let as give an example of the functionality of the modification strategy DFS described just above. Consider the tree $S$ shown in Figure 1(a). The numbers on vertices stem from the DFS visitation ordering. Notice that label $1$ is assigned to a leaf vertex of $S$. The DFS ordering partitions $S$ into the following maximal edge-disjoint paths: $P_1 = \{1, 2, 3, 4\}$, $P_2 = \{2, 5, 6\}$, $P_3 = \{2, 7, 8\}$, $P_4 = \{7, 9, 10\}$.

Then $\mathcal{P}(S) = \{P_1, P_2, P_3, P_4\}$ and $\mathcal{L} = \{1, 2, 3, 4, 2, 5, 6, 2, 7, 8, 7, 9, 10\}$. Suppose that in second stage the realized vertex set is $V' = V \setminus \{2, 7\}$. Then DFS discards all copies of $2$ and $7$ from $\mathcal{L}$, thus producing $\mathcal{L}' = \{1, 3, 4, 5, 6, 8, 9, 10\}$. Subsequently, it traces the ordered $\mathcal{L}$ and adds edges as described previously. The result is shown in Figure 1(b). ∎

**Proposition 1.** PROBABILISTIC STEINER TREE *is **NP**-hard.*




(a) The anticipatory first-stage tree




(b) The modified second stage tree

Figure 1: Application of the DFS modification strategy.

**Proof.** The most technical part of the proof is the inclusion of PROBABILISTIC STEINER TREE in **NPO**[3]. Then, **NP**-hardness of PROBABILISTIC STEINER TREE immediately follows from the fact that setting occurrence-probability 1 for any vertex of the input graph, PROBABILISTIC STEINER TREE coincides with STEINER TREE.

For proving inclusion of PROBABILISTIC STEINER TREE in **NPO** it suffices to prove that the objective function $E(G, S, \texttt{DFS})$ of the problem is computable in polynomial time.

Let $G(V, E)$ be an instance of PROBABILISTIC STEINER TREE and $S$ be an anticipatory Steiner tree on $G$. Suppose that a set $V'$ inducing a (complete) subgraph $G' = G[V']$ of $G$ is presented for optimization. Assume that $S$ is represented by the set of its edges. Assume finally that $S'$ is the Steiner tree computed as described before. Let $V(S)$ be the set of vertices of $S$. Denote by $V'' = V(S')$ the subset of $V'$ consisting of the vertices of $S'$ and by $E''$ the edge-set of $G[V'']$. Let $\mathcal{L}$ be the DFS encoding performed over the anticipatory Steiner tree $S$ by the modification strategy DFS, given as an ordered list of vertices (in the order decided by the DFS) and let $[v_i, v_j]_{\mathcal{L}}$ be the non-empty sublist of $\mathcal{L}$ starting at $v_i$ and ending at $v_j$ (notice that since $\mathcal{L}$ corresponds to a DFS ordering, $i < j$) not including neither $v_i$ nor $v_j$ (it is assumed that if $[v_i, v_j]_{\mathcal{L}} = \emptyset$, or if $i > j$, then $\prod_{v_l \in [v_i, v_j]_{\mathcal{L}}} (1 - p_l) = 0$).

We will prove that the objective value (functional) of $S$ is expressed by:

$$E(G, S, \texttt{DFS}) = \sum_{(v_i, v_j) \in S} w(v_i, v_j)\, p_i p_j + \sum_{(v_i, v_j) \in E'' \setminus S} w(v_i, v_j)\, p_i p_j \prod_{v_l \in [v_i, v_j]_{\mathcal{L}}} (1 - p_l)$$

The expression for $E(G, S, \texttt{DFS})$ consists of two terms, the former expressing the expected cost of surviving edges of the anticipatory tree $S$, and the latter expressing the expected cost of edges added by the modification strategy DFS.

[3]**NPO** is the class of the optimization problems the decision counterparts of which are in **NP**.

For the first term it is straightforward to see that an edge $(v_i, v_j) \in S$ survives if both its endpoints survive, an event occurring with probability $p_i p_j$ (since the two events of survival of each vertex are independent). Hence the expected cost of surviving edges in $S$ is as claimed by the first term of the given expression for $E(G, S, \texttt{DFS})$.

The second term stems by inspection of the functionality of the modification strategy DFS. When the actual second stage graph realizes, vertices missing from the graph are dropped from the initial DFS encoding $\mathcal{L}$, thus producing an encoding $\mathcal{L}' \subseteq \mathcal{L}$. Then, DFS scans $\mathcal{L}'$ and for any pair of consecutive vertices $v_i$ and $v_j$ it connects them by adding $(v_i, v_j)$ if the three following conditions are satisfied: (i) $v_i$, $v_j$ are actually in $\mathcal{L}'$ (ii) $i < j$ (by the DFS-produced labelling) and (iii) $v_i$ is not connected to $v_j$.

Clearly two vertices $v_i, v_j \in \mathcal{L}$ are also in $\mathcal{L}'$ with probability $p_i p_j$. Moreover, an edge $(v_i, v_j)$ is added to the (new) solution $S'$, if $v_i$ is not connected to $v_j$ in second stage. This means (since $v_i$ and $v_j$ are consecutive in $\mathcal{L}'$) that all vertices that existed between $v_i$ and $v_j$ in $\mathcal{L}$ have been dropped in $\mathcal{L}'$. This happens with probability $\prod_{v_l \in [v_i, v_j]_{\mathcal{L}}} (1 - p_l)$. Furthermore, neither $v_i$ nor $v_j$ should also appear as intermediates within $[v_i, v_j]_{\mathcal{L}}$ in the original (first-stage) list encoding $\mathcal{L}$, otherwise (since all intermediate vertices are missing from $\mathcal{L}'$), DFS would not encounter them during its scan of $\mathcal{L}'$. Finally, $[v_i, v_j]_{\mathcal{L}}$ should not be empty, otherwise it would entail a surviving edge between $v_i$ and $v_j$ (recall that $v_i < v_j$ in the encoding), rendering these vertices connected in second-stage. Therefore, the expected cost of added edges is as expressed by the second term of the given expression.

It is easy to see that computation of a single term in the second sum of the functional requires $O(n)$ computations (at most $n + 1$ multiplications). Since we may do this for at most $O(n^2)$ times (the edges in $E$), it follows that the whole complexity of functional's computation is polynomial (in $O(n^3)$). So, PROBABILISTIC STEINER TREE belongs to **NPO** and, with the remark in the beginning of the proof, its **NP**-hardness is also concluded. ∎

Proposition 1 does not derive a compact characterization for the optimal anticipatory solution for PROBABILISTIC STEINER TREE. This is due to the second term of the functional $E(G, S, \texttt{DFS})$ that depends on the structure of the anticipatory solution chosen and of the present subgraph of $G$.

## 2.2 On the approximability of PROBABILISTIC STEINER TREE

We now turn to study approximation of PROBABILISTIC STEINER TREE for several types of anticipatory solutions.

We first show an easy though useful result linking, for any instance $G$ of PROBABILISTIC STEINER TREE, $E^*(G)$ and $\text{opt}(G)$, where $\text{opt}(G)$ is the optimal STEINER TREE-value in $G$ (i.e., the value of an optimal Steiner tree in $G$ by ignoring the vertex-probabilities).

**Fact 1.** $E^*(G) \geqslant \mathrm{opt}(G)$. ▮

Indeed, since $T$ is present in any of the sets $V' \subseteq V$ realized in the second stage, an optimal Steiner tree of $G$ has value smaller than, or equal to, the value of an optimal Steiner tree of any second-stage induced subgraph $G'$ of $G$, i.e., $\mathrm{opt}(G) \leqslant \mathrm{opt}(G')$, for any $G' \subseteq G$. Using it in (2), we get: $E^*(G) \geqslant \sum_{V' \subseteq V} \Pr[V'] \, \mathrm{opt}(G) = \mathrm{opt}(G) \sum_{V' \subseteq V} \Pr[V'] = \mathrm{opt}(G)$, q.e.d.

### 2.2.1 When anticipatory solution is obtained by minimum spanning tree computation

We now turn to the classical minimum-cost spanning tree algorithm on $G[T]$ (sketched in Section 1) and show the following result.

**Proposition 2.** *A minimum-cost spanning tree over the subgraph of $G$ induced by $T$ is a 2-approximation for* PROBABILISTIC STEINER TREE.

**Proof.** Let $H$ be an isomorphic of $G$ with modified edge-weights, i.e., where $(v_i, v_j) \in E$ instead of weight $w(v_i, v_j)$, it has in $H$ weight $w'(v_i, v_j) = p_i p_j w(v_i, v_j)$. Then,

$$E(G, S, \mathtt{DFS}) = m(H, S) = \sum_{(v_i, v_j) \in S} w\left(v_i, v_j\right) p_i p_j \tag{4}$$

On the other hand, since edge-weights in $G$ are at least as large as the corresponding weights in $H$, the following holds (also using Fact 1):

$$E^*(G) \geqslant \mathrm{opt}(G) \geqslant \mathrm{opt}(H) \tag{5}$$

Putting (4) and (5) together and taking into account that $S$ is a 2-approximation for STEINER TREE in $H$ (see Section 1), the result claimed is immediately derived. ▮

A minimum-cost spanning tree $S$ over $G[T]$ (or $H[T]$) is a very particular solution since, thanks to the omnipresence of the vertices of $T$, $S$ remains feasible in second stage, i.e., no completion is needed.

### 2.2.2 More general anticipatory solutions

We so turn to study anticipatory solutions that have not a priori such property (i.e., that need completion in the second stage in order to become feasible Steiner trees) and could contain also non-terminal vertices of $G$, i.e., vertices that have a presence-probability less than 1. More precisely, we consider using an arbitrary $\rho$-approximation algorithm for obtaining a feasible Steiner tree with cost no more than $\rho$ times the cost of an optimal Steiner tree of $G$. We show the following proposition.

**Proposition 3.** *Any $\rho$-approximation algorithm for* STEINER TREE *yields an anticipatory solution resulting in a $2\rho$-approximation for* PROBABILISTIC STEINER TREE. *This ratio is tight even when edge costs are 1 and 2 and an optimal* STEINER TREE-*solution is used as anticipatory (first-stage) solution.*

**Proof.** The proof is based upon the following emphasized claim. *The total cost of edges added to the partial tree $S''$ in order to produce $S'$ is at most $2m(G,S)$, where $S$ is the anticipatory Steiner tree computed in $G$.*

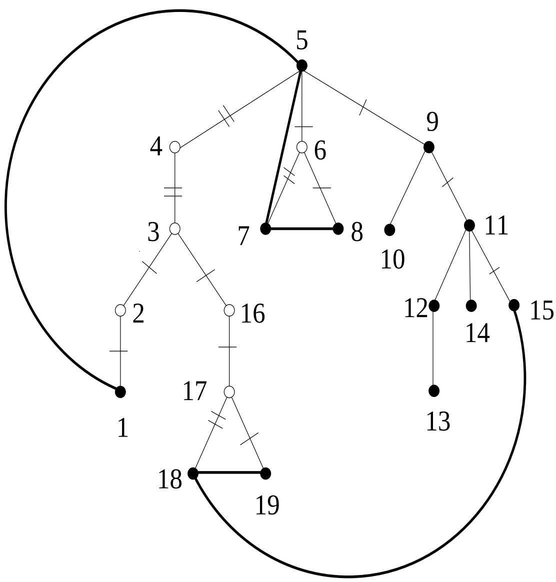



Figure 2: The total cost of edges added to the partial tree $S''$ in order to produce $S'$ (the Steiner tree in $G'$) is at most $2m(G,S)$.

Figure 2, where vertices are named by their DFS ordering, gives an example illustrating the central idea in the proof of the claim that follows. The anticipatory Steiner tree $S$ is the tree with thin lines. We suppose that white vertices are absent in second stage. Hence, the edges added in this stage in order to reconnect $S$ are (the thick) edges $(1,5)$, $(5,7)$, $(7,8)$, $(15,18)$ and $(18,19)$. In the so reconstructed tree $S'$, and since we assume that $G$ is metric, edges $(1,2)$ and $(2,3)$ are counted once, due to the insertion of edge $(1,5)$. Edges $(3,4)$ and $(4,5)$ count twice (due to the insertions of $(1,5)$ and of $(15,18)$. Edges $(5,6)$ and $(6,7)$ count once (for edge $(5,7)$). Edges $(3,16)$, $(16,17)$, $(5,9)$, $(9,11)$, and $(11,15)$ count once, for edge $(15,18)$. Edge $(17,18)$ counts twice, once for edge $(15,18)$ and once for $(18,19)$. Finally, edge $(17,19)$ counts once, for edge $(18,19)$.

In order to formally prove the claim, "rehang", for clarity, the tree $S$ from its leaf numbered by 1 (that becomes the root of $S$). Then, any vertex has a DFS number less than the numbers of the vertices on its subtree.

Consider an edge $(v_i, v_j) \in E$, absent from $G'$ (i.e., either $v_i$ or $v_j$ are absent from $V'$). Then, the anticipatory tree will be completed by an edge $(v_k, v_l)$ such that $k \leqslant i$ and $l \geqslant j$. In other words, $v_k$ is a predecessor of $v_i$ and $v_j$ (or $v_i$ itself) and $v_l$ a successor of them (or $v_j$ itself). Obviously, by the fact that $G$ is metric, $w([v_k, v_l]_{\mathcal{L}}) \leqslant \sum_{(v_r,v_s)\in[v_k,v_l]_{\mathcal{L}}} w(v_r, v_s)$. Hence, the cost of $(v_i, v_j) \in [v_k, v_l]_{\mathcal{L}}$ will count once "forwardly" when $(v_k, v_l)$ is added to $S''$.

Assume that an edge $(v_p, v_q)$, $p \geqslant l$ is added later. Obviously $q \geqslant p \geqslant l$. Edge $(v_i, v_j)$ will count once more "backwardly" if $v_p$ is in the subtree rooted at $v_j$ and $v_q$ is in some subtree rooted at a predecessor of $v_i$ and $v_j$, this subtree being different from the one in which lie $v_i$ and $v_j$. Once more, $w([v_p, v_q]_{\mathcal{L}}) \leqslant \sum_{(v_r,v_s)\in[v_p,v_q]_{\mathcal{L}}} w(v_r, v_s)$.

It is easy to see that edge $(v_i, v_j)$ will not count another time forwardly. Indeed, if it counted once more in this way due, say, to an edge $(v_a, v_b)$, $a < k$ and $b > l$, then, edges $(v_a, v_k)$, $(v_k, v_l)$ and $v_l, v_b)$ should have been added for completion of $S''$ instead of only $(v_a, v_b)$ and $(v_k, v_l)$. Hence $(v_i, v_j)$ would have counted only once.

Assume now that $(v_i, v_j)$ counts a second time backwardly due to an edge $(v_c, v_d)$ added after $(v_p, v_q)$. Obviously, $p < q < c < d$. Then, as mentioned above, $v_c$ is on a subtree rooted at $v_j$. Since $c > q$, there is enough vertices in this subtree in order that one of them takes the DFS number $q$, i.e., $v_q$ should not been located where it has been assumed to be previously (in some subtree rooted at a predecessor of $v_i$ and $v_j$, this subtree being different from the one in which lie $v_i$ and $v_j$). So, $v_i, v_j$ will backwardly count at most once and the claim is proved.

Observe also that the edges of $S$ that will count twice in the completion are absent from $S''$. Indeed, an edge of $S$ that survives in $G'$, so in $S''$, will count in a completion edge only "backwardly".

Denote by $S_1$ the set of edge of $S$ that will count once for evaluating the completion edges and by $S_2$ the (absent) edges of $S$ that will count twice. Denote also by $S'$ the Steiner tree of $G'$ resulting from the completion of $S''$. Then:

$$m\left(G', S'\right) = w\left(S'\right) \leqslant w\left(S''\right) + w\left(S_1\right) + 2w\left(S_2\right) \tag{6}$$

Sets $S_1$ and $S_2$ are obviously disjoint and $S_1 \cup S_2 \subseteq S$. Hence, $w(S_1) + w(S_2) \leqslant w(S) = m(G, S)$. On the other hand, since as noticed just above, $S_2$ is not present in $S''$, $S_2$

and $S''$ are also disjoint, both sets being subsets of $S$. Hence, again $w(S'') + w(S_2) \leqslant m(G, S)$. Putting so these things together with (6), we get: $m(G', S') \leqslant 2m(G, S)$. Taking finally into account that since the vertices of $T$ are, by assumption, always present in $V'$, $\mathrm{opt}(G') \geqslant \mathrm{opt}(G)$, we get from (1):

$$\begin{aligned} E(G, S, \mathtt{DFS}) &\leqslant \sum_{V' \subseteq V} \Pr[V'] \, 2m(G, S) \leqslant 2 \sum_{V' \subseteq V} \Pr[V'] \, m(G, S) \\ &\leqslant 2\rho \sum_{V' \subseteq V} \Pr[V'] \, \mathrm{opt}(G) \leqslant 2\rho \sum_{V' \subseteq V} \Pr[V'] \, \mathrm{opt}(G') \leqslant 2\rho E^*(G) \end{aligned}$$

In order to prove tightness, consider a complete graph $K_{n+1}$ on $n+1$ vertices numbered by $1, 2, \ldots, n+1$. Assume w.l.o.g. that $n \geqslant 6$ is even and that edge $(1, 3)$ and edges $(i, i+1)$, $i = 3, \ldots, n-1$ have cost 2, while the other edges of $K_{n+1}$ have cost 1. Assume, finally, that only vertex 2 is non-terminal and its presence probability is $p_2$. It is easy to see that, in this graph, $\mathrm{opt}(K_{n+1}) = n+1$ and such a tree is realized in several ways and, in particular, by a star $S$ with center 2, or by a path linking somehow the terminals (for example, using edges $(1, 3)$, $(i, i+2)$, for $i$ odd from 1 to $n+1$, edge $(n-2, n+1)$, edges $(j, j-2)$, for $j$ even going from $n-2$ down to 4 and, finally, edge $(4, n)$). Obviously,

$$E^*(K_{n+1}) = p_2(n+1) + (1 - p_2)(n+1) = n+1 \tag{7}$$

Assume now that the anticipatory solution computed is just $S$ (of cost $n+1$) and that the DFS ordering of $S$ has produced the original numbering of $K_{n+1}$, i.e., 1 is the leftmost leaf, 2 the stars'center and $3, \ldots n+1$ the rest of leaves. If 2 is absent, then the completion of the tree will produce the path $(1, 3, 4, \ldots, n+1)$ with cost $2n$. So, the value of $E(K_{n+1}, S, \mathtt{DFS})$ will be:

$$E(K_{n+1}, S, \mathtt{DFS}) = p_2(n+1) + (1 - p_2) \, 2n = 2n - p_2 n + p_2 \tag{8}$$

In Figure 3 an illustration of the discussion just above is provided for $n = 6$. The thick edges of $K_7$ in Figure 3(a) are the ones with cost 2. It is assumed that vertices $1, 3, 4, 5, 6, 7$ are terminals. In Figure 3(b), an optimal Steiner tree of $K_7$ is shown using non-terminal vertex 2. It is assumed that this tree is also the anticipatory solution. Its vertices' numbers represent also their DFS ordering. In Figure 3(c) is shown the DFS completion of $S$ when 2 is absent. Finally, in Figure 3(d), the optimal Steiner tree of $K_7$ using only terminals built as described above is shown.

Dividing (7) by (8) we get a ratio that for small enough values of $p_2$ tends to 2. ∎

# 3 Some comments

By Proposition 3, if for example the $1.55$-approximation algorithm of [30] is used for constructing an anticipatory solution, a $3.1$-approximation for PROBABILISTIC STEINER

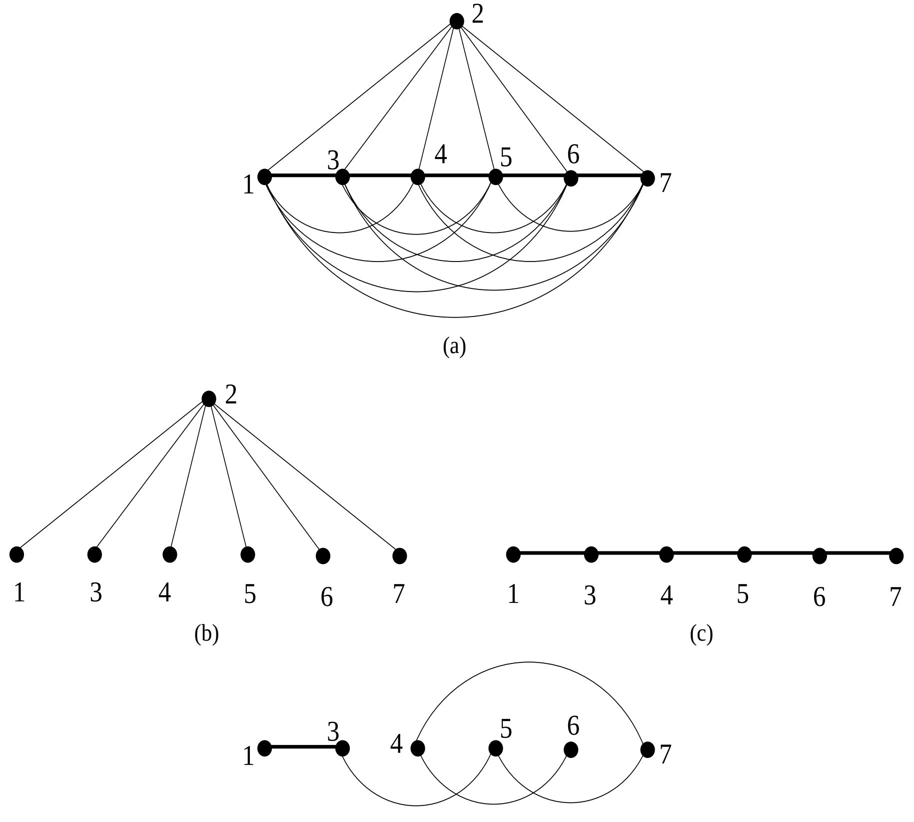



Figure 3: On the tightness of ratio $2\rho$.

TREE is incurred. This result is obviously dominated by Proposition 2. However, Theorem 3 provides a more general structural result linking the approximation of STEINER TREE to the one of PROBABILISTIC STEINER TREE and carries over more general solution structures. This, to our opinion, has its own interest.

But, if one restricts to anticipatory solutions that are optimal for STEINER TREE in the input-graph, then Proposition 3 implies a tight approximation ratio 2 for PROBABILISTIC STEINER TREE. In this case the result incurred is also of practical usefulness since it encompasses more general structures of anticipatory solutions and the approximation ratio obtained is comparable to the one claimed by Proposition 2.

Before concluding the paper let us note that another way for estimating the approximation quality of an anticipatory solution $S$ for PROBABILISTIC STEINER TREE($\mathtt{M}$), for any modification strategy $\mathtt{M}$, is by using the approximation ratio $E(G, S, \mathtt{M})/E(G, S^*, \mathtt{M})$ where, as noticed in Section 1, $S^*$ is an optimal anticipatory solution. By (3), $E^*(G)$ is a lower bound for $E(G, S^*, \mathtt{DFS})$. Henceforth, the approximation results derived in this section remain valid when using approximation ratio $E(G, S, \mathtt{M})/E(G, S^*, \mathtt{M})$ also.

**Acknowledgment.** Many thanks to Cécile Murat for helpful discussions and comments. The comments and suggestions of an anonymous referee are also gratefully acknowledged.

# References

[1] A. Agrawal, P. N. Klein, and R. Ravi. When trees collide: an approximation algorithm for the generalized Steiner problem on networks. *SIAM J. Comput.*, 24(3):440–456, 1995.

[2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The design and analysis of computer algorithms*. Addison-Wesley, Reading, MA, 1975.

[3] I. Averbakh. On the complexity of a class of combinatorial optimization problems with uncertainty. *Math. Programming, Ser. A*, 90:263–272, 2001.

[4] D. J. Bertsimas. *Probabilistic combinatorial optimization problems*. Phd thesis, Operations Research Center, MIT, Cambridge Mass., USA, 1988.

[5] D. J. Bertsimas. The probabilistic minimum spanning tree problem. *Networks*, 20:245–275, 1990.

[6] J. Birge and F. Louveaux. *Introduction to stochastic programming*. Springer, Berlin, 1997.

[7] F. Della Croce, B. Escoffier, C. Murat, and V. Th. Paschos. A priori optimization for minimum graph-coloring in bipartite and split graphs. In O. Gervasi et al., editor, *Proc. International Conference on Computational Science and its Applications, ICCSA'05*, volume 3483 of *Lecture Notes in Computer Science*, pages 202–211. Springer-Verlag, 2005. Full version available at `http://www.lamsade.dauphine.fr/cahiers/PS/cahier218.ps`.

[8] G. W. Dantzig. Linear programming under uncertainty. *Management Sci.*, 1:197–206, 1951.

[9] V. G. Deĭneko and G. J. Woeginger. On the robust assignment problem under a fixed number of cost scenarios. *Operations Research Letters*. To appear.

[10] K. Dhamdhere, V. Goyal, R. Ravi, and M. Singh. How to pay, come what may: approximation algorithms for demand-robust covering problems. In *Proc. FOCS'05*, pages 367–378, 2005.

[11] L. Fleischer, J. Kœnemann, S. Leonardi, and G. Schæfer. Simple cost sharing schemes for multicommodity rent-or-buy and stochastic steiner tree. In *Proc. STOC'06*, pages 663–670, 2006.

[12] C. Gröpl, S. Hougardy, T. Nierhoff, and H. J. Prömel. Approximation algorithms for the Steiner tree problem in graphs. In D.-Z. Du and X. Cheng, editors, *Steiner trees in industry*. Kluwer Academic Publishers, 2000.

[13] A. Gupta, M. Pál, R. Ravi, and A. Sinha. Boosted sampling: approximation algorithms for stochastic optimization. In *Proc. STOC'04*, pages 417–426, 2004.

[14] A. Gupta, M. Pál, R. Ravi, and A. Sinha. What about wednesday? approximation algorithms for multistage stochastic optimization. In *Proc. Approximation Algorithms for Combinatorial Optimization Problems, APPROX'05*, Lecture Notes in Computer Science. Springer-Verlag, 2005.

[15] A. Gupta, R. Ravi, and A. Sinha. An edge in time saves nine: Lp rounding approximation algorithms for stochastic network design. In *Proc. FOCS'04*, pages 218–227, 2004.

[16] N. Immorlica, D. R. Karger, M. Minkoff, and V. S. Mirrokni. On the costs and benefits of procrastination: approximation algorithms for stochastic combinatorial optimization problems. In *Proc. Symposium on Discrete Algorithms, SODA'04*, pages 691–700, 2004.

[17] P. Jaillet. Probabilistic traveling salesman problem. Technical Report 185, Operations Research Center, MIT, Cambridge Mass., USA, 1985.

[18] P. Jaillet. Shortest path problems with node failures. *Networks*, 22:589–605, 1992.

[19] P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*. Kluwer Academic Publishers, Boston, 1997.

[20] E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan, and S. S. Ravi. Algorithmic aspects of topology control problems for ad hoc networks. *Mobile Networks and Applications*, 10(1-2):19–34, 2005.

[21] R. Montemanni, L. M. Gambardella, and A. V. Donati. A branch and bound algorithm for the robust shortest path problem with interval data. *Oper. Res. Lett.*, 32:225–232, 2004.

[22] J. M. Mulvey, R. J. Vanderbei, and S. A. Zenios. Robust optimization of large-scale systems. *Oper. Res.*, 43(2):264–281, 1995.

[23] C. Murat and V. Th. Paschos. The probabilistic longest path problem. *Networks*, 33:207–219, 1999.

[24] C. Murat and V. Th. Paschos. A priori optimization for the probabilistic maximum independent set problem. *Theoret. Comput. Sci.*, 270:561–590, 2002. Preliminary version available at `http://www.lamsade.dauphine.fr/~paschos/documents/c166.pdf`.

[25] C. Murat and V. Th. Paschos. On the probabilistic minimum coloring and minimum $k$-coloring. *Discrete Appl. Math.*, 154:564–586, 2006.

[26] C. Murat and V. Th. Paschos. *Probabilistic combinatorial optimization on graphs*. ISTE and Hermès Science Publishing, London, 2006.

[27] A. Prekopa. *Stochastic programming*. Kluwer Academic Publishers, The Netherlands, 1995.

[28] Stochastic programming community home page. Available at `http://stoprog.org`.

[29] R. Ravi and A. Sinha. Hedging uncertainty: approximation algorithms for stochastic optimization problems. In *Proc. International Conference on Integer Programming and Combinatorial Optimization, IPCO'04*, volume 3064 of *Lecture Notes in Computer Science*, pages 101–115. Springer-Verlag, 2004.

[30] G. Robins and A. Zelikovsky. Improved Steiner tree approximation in graphs. In *Proc. Symposium on Discrete Algorithms, SODA'00*, pages 770–779, 2000.

[31] D. B. Shmoys and C. Swamy. Stochastic optimization is (almost) as easy as deterministic optimization. In *Proc. FOCS'04*, pages 228–237, 2004.

[32] C. Swamy and D. B. Shmoys. Approximation algorithms for 2-stage and multi-stage stochastic optimization. In *Proceedings of the International Conference on Algorithms for Optimization with Incomplete Information*, 2005.

[33] C. Swamy and D. B. Shmoys. Sampling-based approximation algorithms for multi-stage stochastic pptimization. In *Proc. FOCS'05*, pages 357–366, 2005.

[34] V. Vazirani. *Approximation algorithms*. Springer, Berlin, 2001.

[35] P. Wan, K. M. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Mobile Networks and Applications*, 9(2):141–149, 2004.

# La robustesse en recherche opérationnelle et aide à la décision : Une préoccupation multi facettes

Bernard Roy[*]

*Les mathématiques ne sont qu'un moyen de transformation. Seule compte en fait la discussion des prémisses et des résultats (...). En aucun cas, la complexité et la valeur scientifique des déductions ne sauraient donner une valeur scientifique aux prémisses.*

Maurice Allais [1]

**Résumé**

Après avoir précisé dans quel sens j'entends ici le qualificatif robuste, j'explique pourquoi je préfère parler de **préoccupation de robustesse** plutôt que **d'analyse de robustesse** qui risque d'être interprétée dans un sens trop restrictif.

Dans la section 2, je m'efforce d'approfondir et d'expliciter les multiples raisons d'être de cette préoccupation. Pour aller plus avant dans cet examen du caractère multi facettes de la préoccupation de robustesse, je propose, en section 3, de regarder les travaux traitant de robustesse en les répartissant sur trois versants. Dans la section suivante, j'ai recours à ces versants pour montrer à quel point les formes de réponses que devrait susciter cette préoccupation pourraient être encore plus variées qu'elles ne le sont actuellement. Pour terminer, je tente de circonscrire quelques facettes qui mériteraient d'être mieux explorées car susceptibles de porter des pistes de recherche pouvant conduire à d'autres formes de réponses.

**Mots clefs :** Préoccupation de robustesse ; typologie ; critères ; problèmes concrets ; voies de recherche.

---

[*] LAMSADE, Université Paris-Dauphine (roy@lamsade.dauphine.fr).

[1] Le comportement de l'homme rationnel devant le risque : Critique des postulats et axiomes de l'école américaine, *Econometrica*, Volume 21, Number 4, October 1953, 503-546.

**Abstract**

After having specified the meaning of the term “robust” in this paper, I explain why I prefer to talk about **robustness concerns** rather than **robustness analysis**, the latter being likely to be interpreted too narrowly.

In Section 2, I try to deepen the understanding of these concerns and to clarify their numerous *raisons d’être*. To proceed in this investigation of the multiple facets of such robustness concerns, I suggest in Section 3 to look at the works dealing with robustness, distributing them on three territories. In Section 4, I use these territories to show the extent to which the responses to these concerns could be even more varied than they currently are. To conclude, I try to identify several aspects that merit more attention because they could lead to new avenues of research that could in turn yield new forms of responses.

**Key Words:** Robustness concerns; typology; criteria; avenues of research.

# 1. Introduction

En recherche opérationnelle et aide à la décision (RO-AD), il est de plus en plus question de robustesse aussi bien dans les travaux publiés dans les grandes revues scientifiques que dans ceux (souvent moins formels) conduits en entreprise pour traiter un problème spécifique. Ces derniers sont toutefois mal connus car, lorsqu’ils sont publiés (ce qui est relativement rare), ils le sont dans des revues que la communauté scientifique a tendance à ignorer car elle les juge mineures. Le qualificatif « robuste » est employé en RO-AD avec des significations multiples qui prêtent à débat (voir à ce sujet, dans la référence Bulletin (2002-2006), les contributions de Aloulou *et al.* (n° 12, 2005), Dias (n° 13, 2006), Fernandez Barberis (n° 13, 2006), Rios Insua (n° 9, 2004), Rosenhead (n° 6, 2003), Roy (n° 6, 2002), Roy (n° 8, 2003), Sayin (n° 11, 2005), Sevaux, Sörensen (n° 10, 2004), Vincke (n° 8, 2003)). Cette suite de points de vue met en lumière le caractère polysémique de la notion de robustesse, lequel provient notamment du fait que cette notion peut, selon les cas, être apparentée (voire assimilée) à celle de flexibilité, stabilité, sensibilité et même équité.

Dans cet article, j’utilise le terme **robuste** en tant que qualificatif se rapportant à **une aptitude à résister à des « à peu près » ou à des « zones d’ignorances » afin de se protéger d’impacts jugés regrettables, notamment dégradation de propriétés à préserver** (cf. Roy, 2005). Les travaux qui traitent de robustesse cherchent à satisfaire (aussi bien que faire se peut) à cette aptitude. La robustesse relève par conséquent d’une démarche qui répond à une préoccupation : être apte à résister…, à se protéger…

Lors d’une réunion d’un groupe de travail en 2004, Philippe Vincke a fait remarquer que l’expression « analyse de robustesse », bien que fréquemment employée, lui paraissait trop restrictive. J'ai proposé de lui substituer l'expression **préoccupation de robustesse**. En effet, parler d'analyse peut faire penser à un travail qui est effectué *a posteriori*. C'est par exemple ce qui se produit avec une analyse de sensibilité. La robustesse répond au contraire à une préoccupation qui doit intervenir *a priori* dès la formulation même du problème (ceci n'exclut pas de recourir à une analyse de sensibilité pour répondre, le cas échéant, à cette préoccupation).

Dans la prochaine section, je m'efforce d'approfondir et d'expliciter les multiples raisons d'être de cette préoccupation. Pour aller plus avant dans cet examen du caractère multi facettes de la préoccupation de robustesse, je propose, en section 3, de regarder les travaux traitant de robustesse en les répartissant sur trois versants : deux d'entre eux sont définis comme étant clairement opposés ; le troisième, qui se présente comme transversal, est destiné à supporter ceux des travaux qui, par certains traits, se situent plutôt sur l'un des deux premiers et, par d'autres, sur le second. Dans la section suivante, j'ai recours à ces trois versants pour montrer à quel point les formes de réponses que devrait susciter cette préoccupation pourraient être encore plus variées qu'elles ne le sont actuellement. Pour terminer, je tente de circonscrire quelques facettes qui mériteraient d'être mieux explorées car susceptibles de porter des pistes de recherche pouvant conduire à d'autres formes de réponses.

# 2. Raisons d'être de la préoccupation de robustesse

Dans une perspective d'aide à la décision, c'est à mon sens le souhait de tenir compte, autant que faire se peut, de notre ignorance qui est la raison d'être de la préoccupation de robustesse. Dans cette perspective, il ne faut jamais perdre de vue que les décisions que l'on cherche à éclairer seront :

1°) mises à exécution dans un contexte réel qui ne peut pas être rigoureusement conforme au modèle sur lequel l'aide prend appui ;

2°) jugées en faisant référence à un système de valeurs qui paraîtra pertinent (et pas nécessairement stable) dans un futur qui peut ne pas être bien défini, système de valeurs qui, par conséquent, risque de ne pas être tout à fait en accord avec celui qui a été utilisé pour traiter le modèle.

Il y a là deux raisons d'être de manque de conformité et donc d'écart entre :

– d'une part la **représentation formelle** (RF) constituée par le modèle et les procédures de traitement qui lui sont appliquées ;

– d'autre part la **réalité vécue** (RV) dans le cadre de laquelle la ou les décisions prises seront mises à exécution et jugées.

En aide à la décision, il importe donc de chercher à prendre en compte les à peu près, les zones d'ignorances qui sont responsables du fait que la représentation formelle ne peut pas être rigoureusement conforme à la réalité vécue : RF≠RV. Dans cette section, je vais illustrer (sans prétendre à l'exhaustivité) ces à peu près et zones d'ignorances.

Je crois utile auparavant de signaler que, parmi les publications traitant de robustesse, il s'en trouve de plus en plus qui, partant d'une formalisation simple (parfois même simpliste) du rapport susceptible d'exister entre RF et RV, traitent de questions qui relèvent avant tout de préoccupations théoriques. Ces dernières restent malgré tout souvent pertinentes pour la préoccupation de robustesse telle qu'elle a été définie en section 1. Je reviendrai sur cette facette en section 3.

Les à peu près et zones d'ignorances vis-à-vis desquels la préoccupation de robustesse vise à se protéger apparaissent, dans la représentation formelle, sous forme de ce que j'ai proposé d'appeler dans Roy (2005) des **points de fragilité**. La représentation formelle adoptée comme formalisation du problème peut être examinée selon les quatre points de vue ci-après pour mettre en évidence ces points de fragilité.

i) *Façon dont est traitée la mauvaise connaissance* : soit en la négligeant (données traitées comme certaines), soit en la modélisant avec une part d'arbitraire par une distribution de probabilités, des nombres flous, des seuils,…

ii) *Signification préférentielle discutable, voire abusive, attribuée à certaines données* : passage injustifié du qualitatif ou du numérique au quantitatif, signification inappropriée attribuée à des mesures dites objectives, données élaborées dans le cadre d'une procédure de questionnement,…

iii) *Façon de modéliser (notamment en choix de paramètres) utilisée pour prendre en compte un aspect complexe de la réalité difficile à appréhender car imparfaitement défini* : jeux de poids, mesures de capacité, fonctions d'utilité, niveaux de référence ou d'aspiration,…

iv) *Présence de paramètres essentiellement techniques, n'ayant pas ou peu de signification concrète (par exemple imposés par la procédure de traitement)* : écart minimum servant à garantir le caractère strict d'inégalité, bornes servant à délimiter un domaine d'investigation, paramètres intervenant dans le déroulement d'une méta heuristique,…

Se préoccuper de robustesse, c'est, en tout premier lieu, identifier ces points de fragilité dans RF. Ceux-ci dépendent bien évidemment de la façon dont le problème d'aide à la décision a été formulé puis modélisé. Ils peuvent aussi dépendre des procédures de traitement qui vont être utilisées. D'une façon générale, ces points de fragilité apparaissent comme connectés à des sources de contingence, d'incertitude ou d'arbitraire (cf. Roy, 1989 ; Roy, 2005, section 2.2). Les quatre points de vue qui précèdent (croisés avec ces sources qui se situent à un niveau supérieur) sont, je crois, de nature à aider le chercheur opérationnel à inventorier ces points de fragilité lorsqu'il est confronté à un problème concret.

Asseoir cet inventaire en ne s'intéressant qu'à ce qui, dans RF, ressort seulement de ce qui reflète une incertitude peut, dans bien des cas, conduire à laisser de côté un certain nombre de points de fragilité. Le terme « incertitude » ne couvre que très imparfaitement toutes les formes d'à peu près et de zones d'ignorances vis-à-vis desquels il s'agit de résister. C'est le cas par exemple pour les à peu près qui découlent de simplifications, de mauvaises déterminations ou d'options arbitraires. C'est de même le cas pour des zones d'ignorances qui proviennent de mauvaises connaissances relatives à la complexité de phénomènes ou de systèmes de valeurs.

Circonscrire la préoccupation de robustesse à la prise en compte d'incertitudes va généralement de pair avec un mode d'appréhension des relations entre représentation formelle et réalité vécue fondé sur le concept de scénario. Envisagée dans cette optique quelque peu réductrice, la recherche de robustesse repose sur la définition d'une famille finie ou infinie de scénarios. **Cette famille doit permettre d'internaliser dans RF les différentes réalités vécues qui méritent d'être considérées** : c'est l'incertitude dans laquelle on est d'attribuer une vraie valeur à certaines données ou paramètres qui oblige à considérer ces différentes réalités. Chaque scénario est ainsi défini en attribuant une valeur précise à chacune de ces données et paramètres.

J'ai montré dans Roy (2004, 2005) que, pour bien répondre à la raison d'être des préoccupations de robustesse, il était souhaitable de dépasser cette vision réductrice. Si l'on ne veut pas circonscrire la recherche de robustesse à la seule prise en compte d'incertitudes, il convient de s'affranchir du concept de scénario (lequel peut, de plus, être source de malentendus dans certains milieux professionnels). J'ai proposé de le remplacer par celui de **version** de la formulation du problème d'aide à la décision. Chaque version est définie à partir d'une combinaison d'options relatives aux points de fragilité du modèle. Elle représente une réalité méritant d'être considérée. La famille de versions ainsi définie peut en outre, dans certains cas, être insuffisante pour appréhender les relations entre RF et RV. Cela tient au fait que la préoccupation de robustesse peut rendre nécessaire que soit prise en compte non pas une seule procédure de traitement mais toutes celles d'une certaine famille. Les points de fragilité qui rendent nécessaire la prise en compte d'une telle famille peuvent provenir aussi bien de l'existence de paramètres techniques entrant dans la définition des procédures que de la personnalité des experts à qui l'on s'en remet pour effectuer le traitement du modèle (cf. Roy, 2005). Il peut même arriver que la préoccupation de robustesse ne fasse intervenir qu'une seule version de la formulation du problème à laquelle il s'agit d'appliquer la famille de procédures. Dans cette vision élargie de la préoccupation de robustesse, il convient donc de substituer, à la famille de scénarios, une famille constituée de tous les couples (**procédure**, **version**) jugés pertinents. C'est là un aspect de la raison d'être de la préoccupation de robustesse qui n'est pas sans lien avec ce dont il est question dans Vincke (1999a,b).

# 3. Trois versants pour situer les préoccupations de robustesse

Pour pouvoir mettre en évidence (dans la section suivante) les formes multiples que peuvent prendre les réponses à la préoccupation de robustesse compte tenu des raisons d'être qui viennent d'être exposées, je crois éclairant de recourir à une métaphore faisant appel à trois versants sur lesquels il est possible de situer les travaux traitant de robustesse. Dans la présente section, je vais décrire les caractéristiques de ces versants. Les deux premiers, notés *S* (standard) et *C* (concret) présentent des traits caractéristiques qui les opposent à bien des égards. Le troisième, noté *M* (mixte), est destiné à supporter ceux des travaux qui ne présentent que très imparfaitement les traits caractéristiques requis pour pouvoir être positionnés sur l'un des deux versants précédents.

a) *Sur le versant S*, je situe les travaux qui se placent dans le cadre d'une représentation formelle issue de modèles standards de la RO (cf. exemples en section 4) et qui prennent en compte les relations entre cette représentation formelle et des réalités vécues (généralement très peu explicitées) en considérant une famille de scénarios supposée préalablement définie par référence au modèle standard considéré. Ce modèle fait en général intervenir un critère d'optimisation. Le travail porte essentiellement sur la recherche et l'étude de solutions dites robustes eu égard à la famille de scénarios : sont qualifiées de robustes les solutions qui soit optimisent un critère de robustesse (étroitement lié au critère d'optimisation), soit possèdent certaines propriétés exigées (également liées au critère d'optimisation). Lorsque le problème d'optimisation relatif au modèle standard admet un algorithme de résolution polynomial, une des questions souvent abordées consiste à savoir s'il en est de même pour le problème de robustesse.

Parmi beaucoup d'autres, voici quelques publications typiques qui me paraissent pouvoir être situées sur ce versant : Aissi *et al.* (2005a, 2006), Aloulou, Della Croce (2005), Averbackh, Berman (1997), Averbackh, Lebedev (2004, 2005), Ben-Tal, Nemirovski (1999), Bertsimas, Sim (2003, 2004), Briand *et al* (2005), Deineko, Woeginger (to appear), Guttiérrez *et al.* (1996), Hites (2000), Kalaï *et al.* (submitted), Kouvelis, Yu (1997), Montemanni *et al.* (2004), Mulvey *et al.* (1995), Perny *et al.* (to appear), Snyder (to appear), Soyster (1973, 1979), Vallin (1999), Yaman *et al.* (2001), Yu, Yang (1998) (voir également infra les contributions de Aissi *et al*., Gabrel et Murat, Kalaï et Lamboray, Minoux, Paschos *et al.*, Salazar).

b) *Sur le versant C*, je situe les travaux dont le point de départ est un problème concret réel d'aide à la décision qu'il s'agit, en tout premier lieu, de formuler en relation avec des préoccupations de robustesse : la représentation formelle doit ici découler de cette formulation. Les relations à prendre en compte entre RF et RV ne se laissent pas ici nécessairement appréhender au travers d'une famille de scénarios qui s'imposeraient de façon presque évidente. Identifier les points de fragilité, concevoir une famille de versions, éventuellement de couples (procédure, version) font partie intégrante du travail. Celui-ci peut encore porter (comme sur le versant *S*) sur la recherche et l'étude de solutions qualifiées de robustes. Toutefois, la façon de donner sens à ce qualificatif est ici contingente au problème étudié. Celui-ci peut faire intervenir des critères multiples. C'est dire que la robustesse n'est pas nécessairement liée à la présence d'un unique critère d'optimisation. Il peut, de façon plus ou moins complexe, faire intervenir des considérations de flexibilité ou de stabilité. En outre, le concept de solutions robustes peut ne pas être l'objet central du travail : celui-ci peut porter sur la mise en évidence de conclusions qui, selon le cas, pourront être qualifiées de parfaitement robustes, d'approximativement robustes ou de pseudo robustes (cf. Roy, 1998, 2005). Je reviendrai sur cet aspect en section 5.

Les publications qui se rapportent à ce versant sont peu nombreuses car, comme cela est bien connu, les véritables applications ne donnent qu'exceptionnellement lieu à articles acceptés dans les bonnes revues scientifiques. Sans prétendre à l'exhaustivité, je citerai ici Aissi (2005), Aissi *et al.* (2005b), Aloulou, Portmann (2005) ; Carr *et al.* (2006), Chang, Yeh (2002), Durieux (2003), Espinouse *et al.* (2005), Gabrel (1994), Kazakci *et al.* (2006), Rosenblatt, Lee (1987), Roy, Bouyssou (1993), Roy *et al.* (1986), Sevaux, Sörensen (2002, 2004), Siskos, Grigoroudis (2001) Il n'est certainement pas rare que des travaux situés sur le versant *S* ou sur le versant *M* (cf. ci-après) soient exploités dans le cadre d'applications en entreprises (sans pour autant donner lieu à une publication qui se situerait sur ce versant *C*). C'est par exemple le cas des travaux de Bertsimas et Sim qui ont été appliqués à EDF (cf. Remli, 2006) ou encore ceux de Aloulou qui ont été appliqués dans un atelier d'assemblage (cf. Aloulou, Portmann, 2005).

c) *Sur le versant M*, je situe tous les travaux qui ne présentent que trop imparfaitement tous les traits caractéristiques requis pour pouvoir être clairement affectés à l'un des deux versants *S* ou *C*. Cette insuffisance pour une affectation claire à l'un des versants va souvent de paire avec la présence de quelques traits caractéristiques (plus ou moins marqués) relatifs à l'autre versant. Il peut s'agir par exemple de travaux qui, sans avoir pour point de départ un problème réel qu'il est tout d'abord nécessaire de bien identifier, prennent néanmoins appui sur un type de problèmes concrets suffisamment bien défini. Ce type de problème conduit à s'intéresser à une représentation formelle qui s'éloigne des modèles standard. Dans ces conditions, l'identification des points de fragilité, la conception d'une famille de versions ou de couples (procédure, version) ne peuvent alors que découler d'hypothèses énoncées *a priori* sans ancrage dans un cadreconcret bien spécifié. Il peut s'agir aussi de travaux qui portent sur la recherche et l'étude de solutions et/ou de conclusions robustes sans que la représentation formelle ne fasse intervenir un modèle standard de RO particulier. C'est notamment le cas lorsqu'on s'intéresse à une liste finie d'actions potentielles évaluées avec plusieurs critères, voire avec plusieurs juges (et cela sans avoir pour point de départ un problème concret qu'il a fallu commencer par identifier).

A titre d'exemple, je citerai ici Aloulou, Artigues (2006), Beuthe, Scannella (2001), Billaut, Roubellat (1996), Dias *et al.* (2002), Elkhyari *et al.* (2005), Gupta, Rosenhead (1972), Gutiérrez , Kouvelis, (1995), Kouvelis *et al.* (1992), Malcolm, Zenios (1994), Pierreval, Durieux (2006), Rosenhead (2001a,b), Rosenhead *et al.* (1972), Sengupta (1991), Sevaux *et al.* (2005) (voir également infra les contributions de Aloulou et Artigues, Greco *et al.*, Lamboray, Sanlaville, Sörensen et Sevaux, Vallin).

# 4. Variété des formes de réponse auxquelles conduit ou devrait conduire la recherche de robustesse

Ces formes de réponse sont multiples car on peut d'une part imaginer une très grande variété de problèmes théoriques suscitant des travaux sur le versant *S* ou sur le versant *M* et, d'autre part, envisager des contextes concrets extrêmement diversifiés suscitant des travaux trouvant place sur les versants *C* ou *M*.

## 4.1 Sur le versant *S*

La variété des problèmes découle ici des combinaisons qu'offrent quatre sources de diversification.

a) *Le modèle standard*. Il peut en fait s'agir de n'importe lequel des modèles standard de la recherche opérationnelle. Les travaux publiés concernent principalement les suivants (que je cite en vrac sans prétendre à l'exhaustivité) : job shop, flow shop, sac à dos, arbre couvrant, plus court chemin, voyageur de commerce, flot maximum, stable maximum, p-médian et p-centre en localisation,… et, bien sûr aussi, les modèles standard de programmation mathématique (notamment linéaire).

b) *Le (éventuellement les) critère(s) d'optimisation*. Bien qu'un critère unique parfaitement défini aille souvent de paire avec le modèle standard considéré, des variantes ont été étudiées et bien d'autres peuvent être imaginées. C'est par exemple le cas en relation avec le job shop et le flow shop. La prise en compte simultanée de plusieurs critères ne me paraît pas avoir suscité encore beaucoup de travaux, si ce n'est en traitant tous les critères sauf un comme des contraintes.

c) *La famille de scénarios*. Cette famille est généralement définie en considérant la valeur de certains paramètres comme incertaine. Ces paramètres sont soit ceux qui entrent dans la définition du critère d'optimisation, soit ceux qui interviennent dans certaines des contraintes. On suppose que ces paramètres peuvent prendre soit un petit nombre de valeurs, soit toutes celles comprises dans un intervalle. Un scénario est défini en attribuant à chacun des paramètres incertains une des valeurs possibles. On peut considérer ou non que toutes les combinaisons sont autorisées. Dans certains cas, les scénarios sont probabilisés ou ordonnés par vraisemblance décroissante.

d) *La forme des réponses recherchées*. Il s'agit ici principalement d'algorithmes permettant de découvrir ou d'approcher une, ou éventuellement toutes, les solutions qui, relativement à la famille de scénarios peuvent être qualifiées de robustes. Un très grand nombre de ces travaux prennent appui sur l'une des trois façons (que je qualifie précisément ci-après de standard) étudiées dans Kouvelis et Yu (1997) pour appréhender la plus ou moins grande robustesse d'une solution $x$ : sont qualifiées de robustes celles des solutions réalisables qui optimisent un critère $r(x)$ pris comme mesure de la plus ou moins grande robustesse de la solution $x$.

Je rappelle brièvement comment sont définies ces trois mesures standard de la robustesse d'une solution. Elles font intervenir le critère $v$ d'optimisation du modèle standard considéré. Celui-ci attribue à $x$ une valeur $v_s(x)$ dans le scénario $s$. Je suppose ici que optimum signifie maximum.

– *Robustesse absolue*. La mesure de robustesse (qu'il s'agit ici de maximiser) est définie par la valeur que prend la solution dans le pire scénario : $r(x) = \min_s v_s(x)$.

– *Déviation absolue*. La mesure de robustesse (qu'il s'agit ici de minimiser) est définie par la valeur que prend, dans le pire scénario le regret absolu qu'occasionne le fait que la solution diffère de celle qui serait optimale dans ce scénario : $r(x) = \max_s \left[v_s^* - v_s(x)\right]$ ($v_s^*$ : valeur que prend la solution optimale dans le scénario $s$).

– *Déviation relative*. La mesure de robustesse (qu'il s'agit ici de minimiser) est définie par la valeur que prend, dans le pire scénario, le regret relatif qu'occasionne le fait que la solution diffère de celle qui serait optimale dans ce scénario : $r(x) = \max_s \dfrac{v_s^* - v_s(x)}{v_s^*}$.

Certains travaux, notamment parmi les plus récent, sortent du cadre que je viens de présenter[2], en particulier parce qu'ils appréhendent autrement le concept de solution robuste (voir par exemple Bertsimas, Sim, 2003, 2004 ; Beuthe, Scannella, 2001 ; Kalaï, 2006 ; Mulvey *et al.*, 1995 ; Soyster, 1973, 1979).

## 4.2 Sur le versant *C*

Les contextes décisionnels concrets qui suscitent des préoccupations de robustesse sont extrêmement diversifiés. Encore plus diversifiées sont les attentes des décideurs quant à la forme des réponses qu'ils souhaiteraient recevoir de la part de celles et ceux qui cherchent à les aider. Le plus souvent, ils ne sont pas préparés à expliciter les à peu près et zones d'ignorances qui vont compliquer le travail de ces chercheurs et ne savent

[2] Exception faite des travaux cités ci-après, la quasi-totalité des autres citées au 3.a entrent dans ce cadre.

pas nécessairement préciser jusqu'à quel point ils souhaitent éviter certains impacts qu'ils jugent regrettables et voir se dégrader certaines propriétés qu'ils voudraient préserver. Afin de mettre en évidence les facettes multiples de la préoccupation de robustesse qui se présentent sur ce versant *C*, j'aborde successivement ci-après trois grands types de contextes décisionnels et j'illustre dans chacun, à partir de quelques exemples très concrets, ce que peuvent être ces attentes du décideur (sur ce terme, je désigne l'entité pour le compte de qui ou au nom de qui l'aide à la décision s'exerce ; cf. Roy, 1985). Le lecteur trouvera dans Roy (2005, section 3) une série de références bibliographiques se rapportant à ces trois types de contextes décisionnels et notamment aux exemples dont il va être question.

a) *La décision a un caractère ponctuel et exceptionnel (elle ne s'étale ni ne se répète dans le temps)*

i) Choix d'un fournisseur suite à un appel d'offres pour l'acquisition et l'installation d'un nouvel équipement : supposons qu'une quinzaine de réponses aient été reçues et que chacune soit évaluée sur les cinq critères suivants : coût, délai, niveaux de satisfaction relatifs à deux types de performances, confiance accordée au fournisseur quant à son aptitude à respecter les délais et le cahier des charges. Les à peu près et zones d'ignorances affectent ici la façon dont les réponses reçues sont évaluées sur ces cinq critères (plus spécialement sur les trois derniers) ainsi que le rôle qu'il convient de faire jouer à chacun d'eux (autrement dit poids respectif et veto éventuel). Le décideur peut ici attendre du chercheur qu'il lui propose une sélection d'un nombre d'offres aussi restreint que ces à peu près et zones d'ignorances le permettent accompagnées, pour chacune d'elles, des arguments qui justifient qu'elles soient sélectionnées. L'argumentaire doit par exemple permettre de faire comprendre au décideur dans quelles conditions (hypothèses relatives ou à peu près et zones d'ignorances) l'offre considérée s'avère au moins aussi bonne que n'importe quelle autre tout en explicitant les risques qu'elle peut lui faire courir si ces conditions ne sont pas satisfaites.

ii) Fixation des caractéristiques structurelles d'un réseau d'assainissement dans une commune qui en est dépourvue : arrêter de façon optimale la valeur à donner à ces caractéristiques nécessite de connaître de façon suffisamment précise les besoins auxquels il s'agit de satisfaire tout au long de la durée de vie escomptée pour le réseau. Ces besoins sont en fait très mal connus car ils dépendent de multiples facteurs : évolution de la population, de ses habitudes de consommation, implantation de nouvelles activités, évolution de la réglementation des rejets,… Si le chercheur envisage de formuler le problème en termes d'optimisation d'un critère unique, il ne peut restreindre ce critère à la seule prise en compte des coûts prévisionnels de construction et d'entretien du réseau. Il lui faut aussi prendre en compte les coûts que pourront entraîner son adaptation à des besoins qui s'avèreraient impossibles à satisfaire sans modifier les caractéristiques structurelles initialement arrêtées. Il lui faut également prendre en compte les préjudices qu'occasionneraient les surcoûts de construction initiale et d'entretien si les besoins s'avéraient à terme inutilement surestimés. Ceci montre que la formulation d'un critère unique d'optimisation se heurte ici à de sérieuses difficultés. Même si le chercheur opérationnel peut trouver le moyen de les surmonter et de concevoir une famille de scénarios appropriée, il n'est pas évident que cette façon de formuler le problème d'aide à la décision soit la mieux adaptée pour répondre aux attentes du décideur. Attendre en effet que la décision soit robuste, c'est la vouloir telle que, le moment venu, elle puisse être justifiée comme étant une bonne décision et que, sauf circonstances considérées comme imprévisibles, aucun cas de grave insatisfaction des besoins ne se produise. Cet exemple montre que, dans certains cas, la préoccupation de robustesse peut devoir jouer un rôle déterminant dans la formulation même du problème d'aide à la décision.

b) *La décision a un caractère répétitif et/ou séquentiel*

i) Planning d'affectation du personnel navigant aux vols d'une compagnie aérienne : la préoccupation de robustesse incite par exemple à chercher à être apte à faire face à l'indisponibilité non prévue d'équipes (maladie, immobilisation en cours de mission,…) aussi bien qu'à des modifications de plans de vol (avion d'un autre type que celui initialement prévu,…). Comment faire face à ces aléas dans des conditions économiques aussi bien que légales qui soient acceptables sans entraîner pour autant des perturbations de planning susceptibles d'être très mal vécues par le personnel ? Telle est la question à laquelle le chercheur doit s'efforcer de répondre.

ii) Plan d'équipement conçu pour être mis à exécution par étapes : à chaque étape, la préoccupation de robustesse oblige à s'intéresser à la façon dont la décision prise à cette étape affecte les contextes dans lesquels s'inscriront les étapes futures. Cet impact d'une décision prise à l'étape *n* sur ce qui pourra se passer aux étapes ultérieures peut être très mal connu et difficile à formaliser (que l'on songe par exemple au cas d'une usine qui doit être livrée clés en mains puis exploitée sous la responsabilité du constructeur pendant dix ans dans un pays étranger). Il s'agit néanmoins de chercheur à prendre en compte les possibilités d'adaptation et de réaction que la décision prise à chacune des étapes considérées préserve pour les étapes ultérieures. Si mal connues que soient les conditions dans lesquelles les décisions ultérieures pourront être prises, il importe de chercher à arrêter, à chaque étape, une décision qui ne rende pas impossibles ou ne dégrade pas trop les meilleures possibilités de choix ultérieur et, surtout, qui minimise le risque d'enfermer le décideur dans des conditions qui conduiraient à des résultats catastrophiques qui auraient pu être évités. La préoccupation de robustesse s'apparente donc ici à une préoccupation de flexibilité.

c) *La décision porte sur le choix d'une procédure destinée à être utilisée de façon répétitive dans des environnements (lieu, moment,…) dont les caractéristiques peuvent fortement varier*

i) Procédure d'aide à la gestion des réapprovisionnements d'un magasin : considérons le cas où les quantités qui sortent du magasin font l'objet, pour chaque produit, de prévisions portant sur les périodes à venir. Ces prévisions ne sont que des à peu près. Il en est de même pour les délais de livraison annoncés par les fournisseurs. Le gestionnaire attend d'une procédure robuste qu'elle le protège aussi bien des stocks excessifs que de ruptures qui pourraient être jugées économiquement ou psychologiquement inacceptables.

ii) Procédure définissant la façon dont un budget global doit être réparti entre des bénéficiaires dispersés sur un territoire : supposons que ces bénéficiaires soient des entités (par exemple des associations) dont la taille et les besoins sont susceptibles de varier fortement aussi bien au cours du temps que dans l'espace. Face à ces variations, une procédure ne peut être qualifiée de robuste que si elle peut être jugée équitable dans tous les cas où elle doit s'appliquer.

iii) Procédure d'ajustement d'un modèle ayant pour objet de mettre en évidence, à partir d'enquêtes successives, la façon dont divers facteurs, sensés rendre compte d'une réalité, évoluent au cours du temps : pour fixer les idées, supposons que cette réalité ait trait à la satisfaction de consommateurs, de certains biens ou services. Les procédures d'ajustement en question consistent, le plus souvent, à déterminer, suite à chaque enquête, la valeur qu'il convient d'attribuer à ces facteurs pour que le modèle s'ajuste le mieux possible aux résultats de l'enquête. Supposons encore que ce « mieux possible » soit formalisé par un critère d'optimisation qui fait intervenir les écarts entre observation et valeur théorique fournis par le modèle. Les points de fragilité dont la préoccupation de robustesse doit tenir compte proviennent non seulement du fait que les réponses des enquêtés reflètent plus ou moins bien ce que l'on cherche à appréhender mais aussi de certains aspects plus techniques :

– présence dans le modèle de paramètres dépourvus de signification concrète tels que variables d'écart qui doivent être introduites pour pouvoir prendre en compte des inégalités qui doivent être strictes ;

– non unicité de la solution optimale et/ou présence de solutions voisines de l'optimum, toutes ces solutions pouvant être jugées aussi pertinentes pour rendre compte des observations.

La prise en compte de ces aspects techniques pour apprécier la robustesse de la procédure peut être vue ici comme étroitement liée à des préoccupations de stabilité.

## 4.3 Sur le versant *M*

Les formes de réponse auxquelles conduisent ou pourraient conduire les travaux situés sur le versant *M* peuvent être vues comme des formes hybrides de celles que suscitent ou devraient susciter les travaux situés sur les versants *S* et *C*. Les illustrer ici n'apporterait, je crois, rien de plus que ce qui vient d'être dit ci-dessus au 4.1 et au 4.2. Les références citées à la fin du 3.c) fourniront de nouveaux exemples au lecteur qui le souhaite.

# 5. Facettes et pistes de recherche à explorer

Celles et ceux qui sont responsables d'arrêter une décision ou, plus généralement, d'influencer un processus de décision n'attendent pas de l'aide à la décision qu'elle dicte leur conduite. Ils attendent des réponses qui leur apportent des informations utiles servant à baliser leur champ de réflexion et d'action. Il peut s'agir (cf. exemples du 4.b) de propositions de solutions plus ou moins argumentées, de conclusions, notamment du type si … alors …, de systèmes de règles ou de procédures garantissant des propriétés de flexibilité, de stabilité, d'équité,… Je voudrais tout d'abord attirer l'attention sur le fait que le chercheur n'est pas toujours bien armé pour élaborer ces réponses sous une forme adaptée aux attentes.

Quelles que soient les réponses qu'apporte le chercheur, celles-ci ne seront véritablement utiles que si la façon dont elles sont dépendantes ou encore conditionnées par la contingence, l'arbitraire ou encore l'ignorance qui affecte les relations entre d'une part représentation formelle conçue par le chercheur et d'autre part réalité susceptible d'être vécue par le décideur est convenablement prise en compte et explicitée après discussion avec des personnes compétentes. Ceci suppose que la formulation du problème d'aide à la décision soit conçue en tenant compte des préoccupations de robustesse et que les points de fragilité de la représentation formelle (procédure de traitement incluse) soient soigneusement identifiés. Ceci doit permettre d'élaborer ensuite une famille de scénarios ou de versions ou même de couples (procédure, version) qui ne soient ni trop pauvre ni trop riche. Tout ce travail d'élaboration, bien que très fortement conditionné par les particularités aussi bien du problème concret auquel il se rapporte qu'au type de procédures de traitement prises en compte, mériterait de donner lieu à des réflexions méthodologiques se situant sur le versant *M*. Il faudrait pour cela délimiter des classes de problèmes concrets définies en prenant appui d'une part sur une représentation formelle moins simpliste que celle fournie par les modèles standard et, d'autre part, sur une description des caractéristiques de l'environnement concret des problèmes de la classe considérée

Toujours sur le versant *M* à propos de classes bien délimitées de problèmes concrets, il me paraîtrait utile (en vue d'applications effectives) d'élaborer des procédures aptes à séparer ceux des points de fragilité qui ont un impact négligeable de ceux qui, au contraire, paraissent susceptibles d'entraîner une perte de robustesse dommageable. Relativement à ces derniers, il serait intéressant de chercheur à savoir comment procéder pour parvenir à des conclusions parfaitement, approximativement ou pseudo robustes (cf. Roy, 2005, page 44), cela dans l'esprit des exemples qui vont être donnés plus loin.

Je rappelle que, sur le versant *S*, formulation du problème, inventaire des points de fragilité, famille de scénarios ou de versions font partie de ce qui doit être regardé comme des données du problème à propos duquel on se préoccupe de robustesse, données qui sont censées découler d'un travail préalable, lequel a clos une fois pour toutes la question des relations entre RF et RV. Sur ce versant, les réponses actuellement recherchées sont, la plupart du temps, cantonnées dans une mise en évidence de solutions qui peuvent être qualifiées de robustes eu égard à un unique critère d'optimisation, lui aussi préalablement défini. Les quatre sources de diversité présentées au 4.a laissent place, dans ce cadre, pour un très grand nombre de combinaisons susceptibles de donner lieu à publication tout en restant dans les formes de réponses usuelles. Pour achever cet article, je voudrais attirer l'attention sur des pistes de recherche moins classiques qui pourraient conduire à des travaux situés sur le versant *S* ou sur le versant *M* visant à enrichir les formes de réponses et à mieux répondre aux attentes des décideurs, donc susceptibles de nouvelles applications sur le versant *C*.

Chacune des trois mesures standard de robustesse $r(x)$ dont j'ai rappelé la définition au 3.a fait jouer un rôle déterminant aux pires scénarios. Plusieurs auteurs (notamment Kalaï *et al.*, 2005 ; Perny, Spanjaard, 2003 ; Rosenblatt, Lee, 1987 ; Snyder, to appear ; Snyder, Daskin, to appear) ont cherché à atténuer le rôle joué par le pire cas. Ils ont pour cela fait reposer le qualificatif robuste sur le fait de posséder une propriété définie autrement que par l'optimisation d'une mesure de robustesse. Il serait certainement intéressant de chercher à aller plus loin dans cette voie en diversifiant davantage les propriétés requises pour qu'une solution soit qualifiée de robuste.

Ont pourrait aussi s'intéresser à des mesures de robustesse qui contraignent les solutions qui les optimisent à jouir de propriétés remarquables moins simplistes que celles qui caractérisent les trois critères standard rappelés au 4.1d). Tout en m'en inspirant, je propose ci-après trois mesures de robustesse qui me paraissent être de nature à cerner des solutions réellement pertinentes dans de nombreux contextes concrets. Pour introduire ces nouvelles mesures de robustesse, je conserve les notations du 4.1d) en modifiant toutefois (pour des raisons qui apparaîtront ci-après comme naturelles) la définition des $v_s^*$ : dans ce qui suit, $v_s^*$ est la valeur que prend la solution optimale dans le scénario *s* après élimination des solutions qui ne respectent pas la contrainte de borne *b* introduite ci-après (ceci équivaut, pour définir ce maximum, à ne prendre en considération que les solutions dont la mesure de robustesse est non nulle. Afin de simplifier le langage, je continue à faire référence à une famille de « scénarios » bien qu'il serait sans doute plus approprié de parler d'une famille de versions ou même de couples (procédure, version). Les nouvelles mesures proposées ci-après ne présentent un intérêt que si le nombre des « scénarios » (que je suppose fini) n'est pas réduit à quelques unités. C'est notamment le cas lorsque les « scénarios » sont définis en combinant les options retenues (par exemple optimistes, médianes, pessimistes) pour les divers points de fragilité.

Les définitions qui suivent reposent sur la donnée de deux valeurs limites *b* et *w* :

– $b$ ($\geq 0$) sert à caractériser une borne que le décideur demande de chercher à dépasser ou, au contraire, à ne pas franchir (selon celle des trois définitions qui est retenue) dans le plus grand nombre de scénarios possibles ;

– $w$ ($\leq \max_x \min_s (v_s(x))$) est la valeur en dessous de laquelle le décideur refuse de descendre quel que soit le scénario : valeur dite garantie.

Chacune des mesures de robustesse $r_{bw}(x)$ définies ci-après conduit à qualifier de robuste toute solution $x$ qui maximise cette mesure :

i) *(b,w)-robustesse absolue* : $r_{bw}(x) = 0$ si il existe $s$ tel que $v_s(x) < w$ ; sinon, $r_{bw}(x) =$ nombre des scénarios tels que $v_s(x) \geq b$ ; avec $b = \max_x \min_s (v_s(x))$, on retrouve comme solutions robustes celles que le critère standard de robustesse absolue conduit à qualifier de robustes (en effet, pour ces dernières, $r_{bw}(x)$ n'est autre que la cardinalité de la famille de scénarios).

ii) *(b,w)-déviation absolue* : $r_{bw}(x) = 0$ si il existe $s$ tel que $v_s(x) < w$ ; sinon, $r_{bw}(x) =$ nombre des scénarios tels que $v_s^* - v_s(x) \geq b$ ; avec $b = \min_s \max_x \left[ v_s^* - v_s(x) \right]$, on retrouve encore comme solutions robustes celles que le critère standard de déviation absolue conduit à qualifier de robustes lorsqu'on l'applique après élimination des solutions telles que $r_{bw}(x) = 0$.

iii) *(b,w)-déviation relative* : $r_{bw}(x) = 0$ si il existe $s$ tel que $v_s(x) < w$ ; sinon, $r_{bw}(x) =$ nombre des scénarios tels que $\frac{v_s^* - v_s(x)}{v_s^*} \leq b$ ; ici encore, avec $b = \min_s \max_x \frac{v_s^* - v_s(x)}{v_s^*}$, on retrouve comme solutions robustes celles que le critère standard de robustesse relative conduit à qualifier de robustes lorsqu'on l'applique après élimination des solutions telles que $r_{bw}(x) = 0$.

Les solutions ainsi qualifiées de robustes sont celles qui, tout en garantissant une valeur minimum $w$, maximisent le nombre des scénarios permettant d'atteindre une valeur qui :

– est au moins égale à une limite caractérisée par la borne $b$, cela avec la (b,w)-robustesse absolue : ici le décideur, tout en consentant une valeur garantie $w < \max_x \min_s (v_s(x))$, demande de maximiser le nombre des scénarios dans lesquels $v_s(x)$ atteint ou dépasse la borne $b$ ;

– occasionne un regret absolu au plus égal à une limite caractérisée par la borne $b$, cela avec la (b,w)-déviation absolue : ici le décideur, tout en consentant une valeur garantie $w < \max_x \min_s(v_s(x))$, demande de maximiser le nombre des scénarios dans lesquels le regret absolu reste au plus égal à la borne *b* ;

– occasionne un regret relatif au plus égal à une limite caractérisée par la borne *b*, cela avec la (b,w)-déviation relative : ici le décideur, tout en consentant une valeur garantie $w < \max_x \min_s(v_s(x))$, demande de maximiser le nombre des scénarios dans lesquels le regret relatif reste au plus égal à la borne *b*.

A chaque décision ainsi qualifiée de robuste se trouve associé un petit nombre de scénarios qui ne respectent pas la borne imposée. Connaître ces scénarios peut constituer une information très utile au décideur pour choisir parmi les diverses solutions robustes. Il peut également souhaiter connaître la façon dont cet ensemble de scénarios varie lorsqu'il modifie quelque peu les valeurs de *b* et *w*.

Le lecteur trouvera en annexe un exemple numérique élémentaire conçu pour illustrer l'intérêt de la (b,w)-robustesse absolue et de la (b,w)-déviation absolue.

Dans le cas où chaque scénario peut être affecté d'une probabilité, on peut, dans chacune des trois définitions de $r_{bw}(x)$ introduites ci-dessus, substituer au nombre des scénarios compatibles avec la borne *b* la somme de leurs probabilités. Lorsque le nombre des scénarios est très grand, $r_{bw}(x)$ peut être défini non pas par le nombre mais par la proportion de ceux qui respectent la borne imposée. Dans tous les cas, discuter avec le décideur des valeurs qu'il souhaite attribuer aux paramètres *b* et *w* constitue un moyen pour appréhender son attitude face au risque que lui fait courir la présence des à peu près et zones d'ignorances.

Il serait je crois intéressant de trouver, pour chacune des trois mesures de robustesse proposées ci-dessus et relativement à certaines classes de problèmes bien identifiées, des façons de procéder pour asseoir des conclusions robustes telles que les suivantes.

*Conclusions parfaitement robustes* : avec *b*=… et *w*=…, quelle que soit la solution robuste, le nombre (ou la proportion) des scénarios qui ne respectent pas la borne imposée est au moins égal (ou au plus égal) à…

*Conclusions approximativement robustes* : avec *b*=… et *w*=…, la solution *x* est robuste et les scénarios qui ne respectent pas la borne imposée, bien que difficiles à caractériser rigoureusement, résultent d'une combinaison d'options extrêmes ayant trait aux points de fragilité suivants…

*Conclusions pseudo robustes* : une métaheuristique a permis de mettre en évidence le fait que, avec *b*=… et *w*=…, il ne doit pas exister de solutions robustes qui respectent la borne imposée pour un nombre (ou une proportion) de scénarios supérieur à … et, parmi ceux qui ne respectent pas cette borne, on trouve presque toujours les scénarios suivants…

Jusqu'ici, je me suis placé dans le cas où il existe un unique critère à optimiser et où le sens donné au qualificatif robuste repose sur la définition d'une mesure de robustesse qui fait intervenir ce critère unique. Il conviendrait de s'intéresser aussi à des classes de problèmes bien identifiées dans lesquelles les solutions sont jugées selon plusieurs critères. Dans ces nouveaux cas, prendre en compte la famille de critères pour concevoir une mesure de robustesse est une voie qui mérite d'être explorée. Il peut cependant être intéressant de chercher à asseoir le qualificatif robuste sur des propriétés requises qui ne reposent pas sur l'optimisation d'une mesure de robustesse (notamment parce qu'elles en font intervenir plusieurs). Cette façon de traiter la préoccupation de robustesse me paraît être tout à fait pertinente lorsque l'on est en présence d'un ensemble fini d'actions potentielles, le qualificatif robuste devant alors s'appliquer soit à une action sélectionnée ou à une procédure de sélection, à un classement des actions potentielles ou à une procédure de rangement, soit encore à une affectation des actions potentielles à des catégories prédéfinies ou à une procédure de tri. Ce sont là d'autres pistes qui commencent à être explorées (voir infra Greco *et al.*, Lamboray).

# Remerciements

Une première version de cet article a bénéficié des remarques et commentaires de Hassene Aissi et de Daniel Vanderpooten. Ensuite, Mohamed Ali Aloulou et Eric Sanlaville ont bien voulu examiner une seconde version et me faire part de diverses critiques et suggestions très pertinentes. J'adresse ici mes plus vifs remerciements aux uns comme aux autres. Je suis également très reconnaissant envers Dominique François qui, une fois de plus, a bien voulu assurer avec beaucoup de soin les frappes et corrections successives en attirant mon attention sur les moindres anomalies ou incorrections de style qu'elle pouvait détecter.

# Références

Aissi H. (2005) : *Approximation et résolution des versions min-max et min-max regret de problèmes d'optimisation combinatoire*, Université Paris-Dauphine, Thèse de Doctorat.

Aissi H., Bazgan C., Vanderpooten D. (2005a) : "Complexity of the min-max and min-max regret assignment problem", *Operations Research Letters* 33, 6, 634-640.

Aissi H., Vanderpooten D., Vanpeperstraete J.M. (2005b) : "Robust approaches for the data association problem", in: *Proceedings of the 8th International Conference on Information Fusion (FUSION 2005)*, Philadelphia, USA.

Aissi H., Bazgan C., Vanderpooten D. (2006) : “Approximation of min-max and min-max regret versions of some combinatorial optimization problems”, *European Journal of Operational Research*.

Aloulou M.A., Artigues C. (2006) : “Worst-case evaluation of flexible solutions in disjunctive scheduling problems”, in: *The 2006 International Conference on Compputational Science and its applications (ICCSA 2006)*, Glasgow, UK, May 8-11, 2006.

Aloulou M.A., Della Croce F. (2005) : “On the complexity of single machine scheduling problems under scenario-based uncertainty”, Université Paris-Dauphine, *Annales du LAMSADE* 4-5, 1-14.

Aloulou M.A., Portmann M.C. (2005) : « Une approche proactive réactive efficace, cas d’un atelier d’asssemblage », in : Billaut J.C., Moukrim A., Sanlaville E. (sous la direction de) : *Flexibilité et robustesse en ordonnancement*, Paris, Lavoisier, 243-261.

Averbakh I., Berman O. (1997) : “Minimax regret p-center location on a network with demand uncertainty”, *Location Science* 5, 4, 247-254.

Averbakh I., Lebedev V. (2004) : “Interval data minmax regret network optimization problems”, *Discrete Applied Mathematics* 138, 289-301.

Averbakh I., Lebedev V. (2005) : “On the complexity of minmax regret linear programming”, *European Journal of Operational Research* 160, 1, 227-231.

Ben-Tal A., Nemirovski A. (1999) : “Robust solutions of uncertainty linear program”, *Operations Research Letters*, Volume 25, Issue 1, 1-13.

Bertsimas D., Sim M. (2003) : “Robust discrete optimization and network flows”, *Mathematical Programming* B, 98, 49-71.

Bertsimas D., Sim M. (2004) : “The price of robustness”, *Oper. Res.* 52, 1, 35-53.

Beuthe M., Scannella G. (2001) : “Comparative analysis of UTA multicriteria methods”, *European Journal of Operational Research*, vol. 130, n° 2, 246-262.

Billaut J.C., Moukrim A., Sanlaville E. (sous la direction de) (2005) : *Flexibilité et robustesse en ordonnancement*, Paris, Lavoisier.

Billaut J.C., Roubellat F. (1996) : “A new method for workshop real time scheduling”, *International Journal of Production Research* 34, 6, 1555-1579.

Briand C., Huguet M.J., La H.T., Lopez P. (2005) : Approches par contraintes pour l'ordonnancement robuste », in : Billaut J.C., Moukrim A., Sanlaville E. (sous la direction de) : *Flexibilité et robustesse en ordonnancement*, Paris, Lavoisier, 191-218.

Bulletin (2002-2006) du Groupe de Travail Européen « Aide Multicritère à la Décision » n° 6 à 13 (disponible à : http://www.inescc.pt/~ewgmcda/Articles.html).

Carr R.D., Greenberg H.J., Hart W.E., Lauer K.E., Lin H., Morrison T., Phillips C.A. (2006) : "Robust optimisation of contaminant sensor placement for community water system", *Mathematical Programming*, Serie B, 106(1-2), 337-356.

Chang Y.H., Yeh C.H. (2002) : "A survey analysis of service quality for domestic airlines", *European Journal of Operational Research* 139, 166-177.

Deineko V.G., Woeginger G.J. (to appear) : "On the robust assignment problem under a fixed number of cost scenarios", *Operations Research Letters*.

Dias L.C., Mousseau V., Figueira J., Clímaco J. (2002) : "An aggregation/disaggregation approach to obtain robust conclusions with ELECTRE TRI", *European Journal of Operational Research*, vol. 138, n° 2, 332-348.

Durieux S. (2003) : *Analyse de performance par simulation et conception de systèmes de production robustes*, Université Blaise Pascal Clermont II, Thèse de Doctorat.

Elkhyari A., Guéret C., Jussien N. (2005) : « Ordonnancement dynamique de projet », in : Billaut J.C., Moukrim A., Sanlaville E. (sous la direction de) : *Flexibilité et robustesse en ordonnancement*, Paris, Lavoisier, 325-344.

Escudero L.F. (1994) : "Robust decision making as a decision making aid under uncertainty", in: Rios S. (ed.): *Decision Theory and Decision Analysis: Trends and Challenges*, Kluwer Academic Publishers, 126-138.

Espinousse M.L., Jacomino M., Rossi A. (2005) : « Robustesse de la configuration d'un parc de machines », in : Billaut J.C., Moukrim A., Sanlaville E. (sous la direction de) : *Flexibilité et robustesse en ordonnancement*, Paris, Lavoisier, 51-59.

Gabrel V. (1994) : *Méthodologie pour la planification de production de systèmes d'observation de la terre par satellites – Aspects algorithmiques et multicritères*, Université Paris-Dauphine, Thèse de Doctorat.

Gupta S.K., Rosenhead J. (1972) : "Robustness in sequential investment decision", *Management Science* 15, 2, 18-29.

Gutiérrez G.J., Kouvelis P. (1995) : "A robustness approach to international sourcing", *Annals of Operations Research* 59, 165-193.

Gutiérrez G.J., Kouvelis P., Kurawarwala S.A. (1996) : « A robustness approach to uncapacitated network design problem », *European Journal of Operational Research* 94, 2, 362-376.

Hites R. (2000) : *The robust shortest path problem*, PhD Thesis, Université Libre de Bruxelles, Bruxelles, Belgique.

Kalaï R. (2006) : *Une nouvelle approche de robustesse : Application à quelques problèmes d'optimisation*, Université Paris-Dauphine, Thèse de Doctorat.

Kalaï R., Aloulou M.A., Vallin P., Vanderpooten D. (2005) : "Robust 1-median location problem on a tree", in: *Proceedings of the third ORP3 conference (a EURO Conference for Young OR Researchers and Practitioners)*, Valencia, Spain, 201-212.

Kalaï R., Aloulou M.A., Vallin P., Vanderpooten D. (submitted) : "Lexicographic α-robustness: an application to the 1-median problem", *European Journal of Operational Research*.

Kazakci A., Rozakis S., Vanderpooten D. (2006) : "Energy crop supply in France: a min-max regret approach", *Journal of the Operational Research Society* (to appear).

Kouvelis P., Yu G. (1997) : *Robust discrete optimization and its applications*, Kluwer Academic Publishers.

Kouvelis P., Kurawarwala A.A., Gutiérrez G.J. (1992) : "Algorithms for robust single and multiple period layout planning for manufacturing systems", *European Journal of Operational Research* 63, 2, 287-303.

Malcolm S.A., Zenios S.A. (1994) : "Robust optimization for power systems capacity expansion under uncertainty", *Journal of the Operational Research Society* 45, 1040-1049.

Montemanni R., Gambardella L.M., Donati A.V. (2004) : "A branch and bound algorithm for the robust shortest path problem with interval data", *Operations Research Letters* 32, 225-232.

Mulvey J.M., Vanderbei R.J., Zenios S.A. (1995) : "Robust optimization of large-scale systems", *Operations Research Letters* 43, 2, 264-281.

Perny P., Spanjaard O. (2003) : "An axiomatic approach to robustness in search problems with multiple scenarios", in: *Proceedings of the19th Conference on Uncertainty in Artificial Intelligence*, Acapulco, Mexico, 469-476.

Perny P., Spanjaard O., Storme L.X. (to appear) : "Enumeration of robust paths and spanning trees in multivalued graphs", *Annals of Operations Research*.

Pierreval H., Durieux S. (2006) : “Robust simulation with a base environmental scenario”, *European Journal of Operational Research*.

Remli N. (2006) : *Gestion de stockages gaziers*, Université Paris-Dauphine, Mémoire du Master Modélisation, Optimisation, Décision et Organisation.

Rosenblatt M.J., Lee H.L. (1987) : “A robustness approach to facilities design”, *International Journal of Production Research* 25, 4, 479-486.

Rosenhead J. (2001a) : “Robustness analysis: keeping your options open”, in: Rosenhead J., Mingers J. (eds.): *Rational analysis for a problematic world revisited: problem structuring methods for complexity, uncertainty and conflicts*, Chichester, Wiley, 181-207.

Rosenhead J. (2001b) : “Robustness to the first degree”, ”, in: Rosenhead J., Mingers J. (eds.): *Rational analysis for a problematic world revisited: problem structuring methods for complexity, uncertainty and conflicts*, Chichester, Wiley, 209-222.

Rosenhead J., Elton M., Gupta S.K. (1972) : “Robustness and optimality criteria for strategic decisions”, *Operational Research Quarterly* 23, 4, 413-423.

Roy B. (1985) : *Méthodologie multicritère d’aide à la décision*, Paris, Economica.

Roy B. (1989) : “Main sources of inaccurate determination, uncertainty and imprecision in decision models”, *Mathematical and Computer Modelling*, vol. 12, n° 10/11, 1245-1254.

Roy B. (1998) : “A missing link in OR-DA, Robustness analysis”, *Foundations of Computer and Decision Sciences*, vol. 23, n° 3, 141-160 (version française : “Une lacune en RO-AD : les conclusions robustes, Université Paris-Dauphine, *Cahier du LAMSADE* n° 144, 1997).

Roy B. (2004) : « Robustesse de quoi et vis-à-vis de quoi mais aussi robustesse pourquoi en aide à la décision ? », in : Henggeler Antunes C., Figueira J., Clímaco J. (eds.) : *Aide Multicritère à la Décision – Multiple Criteria Decision Aiding*, 56e Journées du Groupe de Travail Européen « Aide Multicritère à la Décision » – 56th Meeting of the European Working Group « Multiple Criteria Decison Aiding », Coimbra, Portugal, October 3-5, 2002, Comissão de Coordenação e Desenvolvimento Regional do Centro, INESC Coimbra, Faculdade de Economia da Universidade de Coimbra, 29-39.

Roy B. (2005) : « A propos de robustesse en recherche opérationnelle et aide à la décision », in : Billaut J.C., Moukrim A., Sanlaville E. (sous la direction de) : *Flexibilité et robustesse en ordonnancement*, Paris, Lavoisier, 35-50.

Roy B., Bouyssou D. (1993) : *Aide multicritère à la décision – Méthodes et cas*, Paris, Economica.

Roy B., Présent M., Silhol D. (1986) : “A programming method for determining which Paris metro station should be renovated”, *European Journal of Operational Research* 24, 318-334.

Sengupta J.K. (1991) : “Robust decisions in economics models”, *Computers and Operations Research* 18, 2, 221-232.

Sevaux M., Sörensen K. (2002) : « Genetic algorithm for robust schedules », in: *Proceedings of the 8th International Workshop on Project Management and Scheduling (PMS’2002)*, Valencia, Spain, 330-333.

Sevaux M., Sörensen K. (2004) : “A genetic algorithm for robust scheduling in a non-machine environment with ready times and due dates”, *4OR* 2, 129-147.

Sevaux M., Sörensen K., Le Quéré Y. (2005) : « Métaheuristiques pour la planification et l’ordonnancement robuste », in : Billaut J.C., Moukrim A., Sanlaville E. (sous la direction de) : *Flexibilité et robustesse en ordonnancement*, Paris, Lavoisier, 115-133.

Siskos Y., Grigoroudis E. (2001) : “Measuring customer satisfaction for various services”, in : Bouyssou D., Jacquet-Lagrèze E., Perny P., Slowinski R., Vanderpooten D., Vincke P. (eds.) : *Aiding Decisions with Multiple Criteria – Essays in Honor of Bernard Roy*, Kluwer Academic Publishers, 457-482.

Snyder L.V. (to appear) : “Facility location under uncertainty: A review”, *IIE Transactions*.

Snyder L.V., Daskin M.S. (to appear) : “Stochastic p-robust location problem”, *IIE Transactions*.

Soyster A.L. (1973) : “Convex programming with set-inclusive constraints and applications to inexact linear programming”, *Oper. Res.* 21, 1154-1157.

Soyster A.L. (1979) : “Inexact linear programming with generalized resource sets”, *European Journal of Operational Research* 3, 316-321.

Spanjaard O. (20203) : *Exploitation de préférences non-classiques dans les problèmes combinatoires : modèles et algorithmes pour les graphes*, Université Paris-Dauphine, Thèse de Doctorat.

Vallin P. (1999) : « Détermination d'une période économique robuste dans le cadre du modèle de Wilson », *RAIRO Recherche Opérationnelle*, vol. 33, n° 1, 47-67.

Vincke P. (1999a) : "Robust and neutral methods for aggregating preferences into an outranking relation", *European Journal of Operational Research*, vol. 112, n° 2, 405-412.

Vincke P. (1999b) : "Robust solutions and methods in decision-aid", *Journal of Multi-Criteria Decision Analysis*, vol. 8, 181-187.

Yaman H., Karasan O.E., Pinar M.C. (2001) : "The robust spanning tree problem with interval data", *Operations Research Letters* 29, 31-40.

Yu G.J., Yang J. (1998) : "On the robust shortest path problem", *Computers and Operations Research* 25, 457-468.

# ANNEXE : EXEMPLE NUMÉRIQUE

On considère ici cinq actions évaluées dans vingt scénarios (cf. tableau 1) selon un critère *v* servant à exprimer un gain.

**Tableau 1**

| Actions | $s_1$ | $s_2$ | $s_3 \ldots s_{10}$ | $s_{11} \ldots s_{20}$ |
|---|---|---|---|---|
| $x_1$ | 150 | 180 | 180 | 180 |
| $x_2$ | 170 | 140 | 170 | 180 |
| $x_3$ | 180 | 130 | 180 | 190 |
| $x_4$ | 200 | 120 | 200 | 210 |
| $x_5$ | 90 | 90 | 220 | 210 |

**AVEC LA (b,w)-ROBUSTESSE ABSOLUE** : Rappelons que les solutions robustes sont ici celles qui, tout en garantissant le gain *w*, maximisent le nombre des scénarios qui procurent un gain au moins égal à *b*.

1°) *Avec un gain garanti w=120* : $x_5$ est éliminée (il en est ainsi dès que *w*>90).

a) Si *b*>200, alors il n'existe aucune solution robuste.

b) Si *b*=200, alors $x_4$ est la seule solution robuste et 19 scénarios sur 20 procurent un gain au moins égal à 200 ; ce résultat est inchangé si le décideur se contente de *b*>180.

c) Si *b*=180, alors $x_4$ reste robuste ; $x_1$ et $x_3$ le deviennent (ces deux nouvelles solutions garantissent un gain minimum strictement supérieur à 120). Ce résultat est inchangé si le décideur se contente de *b*>170.

d) Si *b*=170, alors les quatre solutions sont robustes.

2°) *Avec un gain garanti w=130* : $x_5$ mais aussi $x_4$ sont éliminées.

a) Si *b*>190, alors il n'existe aucune solution robuste.

b) Si *b*=190, alors $x_3$ est la seule solution robuste et 10 scénarios sur 20 procurent un gain au moins égal à 190 ; ce résultat est inchangé si le décideur se contente de *b*>180.

c) Si $b$=180, alors $x_3$ reste robuste et $x_1$ le devient (cette nouvelle solution garantit un gain minimum >130) ; ce résultat reste inchangé si de décideur se contente de $b$>170.

d) Si $b$=170, alors les trois solutions sont robustes.

Ces premiers résultats contribuent à mettre en évidence le caractère éminemment subjectif de la notion de **solution robuste**. Ils soulignent la nécessité de pouvoir engager un dialogue avec le « décideur » (celui pour le compte de qui ou au nom de qui l'aide à la décision s'exerce) afin de lui permettre de percevoir les conséquences de ses *exigences et/ou souhaits*. Les résultats qui précèdent constituent des conclusions parfaitement robustes qui, **dans l'hypothèse où le scénario $s_2$ n'est pas jugé plus vraisemblable que l'un quelconque des 19 autres**, justifient les recommandations suivantes :

– si tout gain inférieur à 120 est jugé inacceptable, alors choisir $x_4$, ce qui procurera un gain égal à 200 dans tous les scénarios autres que $s_2$ ;

– si tout gain inférieur à 130 est jugé inacceptable, alors choisir $x_3$, ce qui procurera un gain égal à 190 dans 10 scénarios sur 20 et au moins égal à 180 dans 19 scénarios sur 20.

**AVEC LA (b,w)-DÉVIATION ABSOLUE** : Rappelons que les solutions robustes sont ici celles qui, tout en garantissant le gain $w$, maximisent le nombre des scénarios qui limitent le regret (vis-à-vis de la solution optimale dans le scénario) à la valeur $b$.

1°) *Avec w=120* : $x_5$ est éliminée.

a) Si $0 \leq b < 20$, alors $x_4$ est seule robuste : elle conduit à un regret nul dans tous les scénarios autres que $s_2$ ; faisons observer que, avec cette définition de la robustesse, il y a toujours des solutions robustes avec $b$=0 (et, *a fortiori*, avec $b$>0). Toutefois, le nombre des scénarios dans lesquels le regret est nul peut être réduit à un seul.

b) Si $20 \leq b < 30$, alors $x_4$ reste robuste et $x_3$ le devient. Le regret reste inférieur à 30 dans tous les scénarios autres que $s_2$ et, dans ce scénario, le regret est moindre avec $x_3$ qu'avec $x_4$.

c) Si $30 \leq b < 40$, alors $x_4$ et $x_3$ restent robustes et $x_1$ le devient ; on remarquera que, dans 10 des 18 scénarios autres que $s_1$ et $s_2$, $x_1$ conduit à un regret strictement supérieur à celui auquel conduit $x_3$ (et, *a fortiori*, $x_4$).

d) Si $40 \leq b < 50$, alors $x_1$, $x_3$ et $x_4$ ne sont plus robustes mais $x_2$ le devient : cette solution conduit en effet à un regret qui reste inférieur à 50 quel que soit le scénario alors que, avec $x_1$, $x_3$ ou $x_4$, ce regret atteint ou dépasse 50 dans l'un des 20 scénarios.

e) Si $50 \leq b < 60$, $x_2$ reste robuste et $x_1$ et $x_3$ le redeviennent, ce qui n'est pas le cas de $x_4$ qui conduit à un regret de 60 dans le scénario $s_2$.

2°) *Avec w=130* : $x_4$ et $x_5$ sont éliminées.

a) Si $0 \leq b < 10$, alors $x_3$ est seule robuste ; elle conduit à un regret nul dans tous les scénarios autres que $s_2$.

b) Si $b \geq 10$, alors $x_1$, $x_2$, $x_3$ sont robustes.

Cette seconde définition de la robustesse conduit encore à recommander $x_4$ si *w*=120 et $x_3$ si *w*=130. Elle a par ailleurs mis en évidence le fait qu'une solution robuste pour certaines valeurs de *b* peut cesser de l'être pour une valeur $b' > b$. Ceci n'est nullement choquant comme le montre le 1°)d) ci-dessus. Le fait d'être moins exigeant sur la limitation des regrets peut faire apparaître des solutions dans lesquelles le nombre des scénarios qui violent la nouvelle limite $b'$ est strictement plus faible que celui des scénarios qui violait la limite *b* dans la solution qui n'est plus robuste avec la borne $b'$.Telle solution qui, dans ces conditions, cesse d'être robuste peut le redevenir (cf. 1.e) ci-dessus) lorsque l'on est encore moins exigeant sur la limitation des regrets avec une borne $b'' > b'$ qui correspond précisément à la présence de cette valeur du regret dans chacun des scénarios qui avaient entraîné l'élimination de cette solution. Le même phénomène peut se produire avec la (b,w)-robustesse absolue.

Pour terminer, il me paraît intéressant de comparer les résultats précédents avec ceux auxquels conduisent *β*-robustesse et l'*α*-robustesse lexicographique. Le lecteur trouvera la définition de ces deux façons d'appréhender la robustesse dans l'article de Kalaï et Lamboray qui figure dans le présent volume.

**AVEC LA β-ROBUSTESSE**

1°) *En l'absence de* $x_5$

a) Si *β*<40, alors il n'existe pas de solution robuste.

b) Si $40 \leq \beta < 50$, alors $x_2$ est la seule solution robuste.

c) Si $50 \leq \beta < 60$, alors $x_1$, $x_2$ et $x_3$ sont robustes.

d) Si $\beta \geq 60$, les quatre solutions sont robustes.

2°) *En présence de* $x_5$

a) Si $\beta < 50$, alors il n'existe pas de solution robuste.

b) Si $50 \leq \beta < 60$, alors $x_1$, $x_2$ et $x_3$ sont robustes.

c) Si $\beta \geq 60$, alors $x_1$, $x_2$, $x_3$ et $x_4$ sont robustes.

**AVEC L'$\alpha$-ROBUSTESSE LEXICOGRAPHIQUE**

1°) *En l'absence de* $x_5$

a) Si $\alpha < 20$, alors il n'existe pas de solution robuste.

b) Si $20 \leq \alpha < 30$, $x_3$ est la seule solution robuste.

c) Si $\alpha \geq 30$, alors les quatre solutions sont robustes.

2°) *En présence de* $x_5$

a) Si $\alpha < 30$, alors il n'existe pas de solution robuste.

b) Si $30 \leq \alpha < 40$, alors $x_3$ et $x_4$ sont les seules solutions robustes.

c) Si $\alpha \geq 40$, alors $x_1$, $x_2$, $x_3$ et $x_4$ sont robustes.

On observera que, avec ces deux dernières façons d'appréhender la robustesse, l'intérêt de la solution $x_4$ n'est pas mis en évidence : tout au plus, sous réserve que $x_5$ soit présente, l'$\alpha$-robustesse lexicographique fait apparaître $x_4$ comme robuste au même titre que $x_3$. Cela n'a rien de surprenant. En effet, la $\beta$-robustesse et l'$\alpha$-robustesse lexicographique font jouer un rôle important aux résultats les plus mauvais. Ce rôle est certes moins déterminant qu'il ne l'est dans la forme standard de la déviation absolue (cf. 4.1d)) : les paramètres $\alpha$ et $\beta$ permettent d'en atténuer la portée. Ces paramètres, qui doivent servir à asseoir le dialogue avec le décideur, ne lui permettent pas d'exprimer le moindre souhait quant à l'espoir d'obtenir de bons résultats. Ce type de souhaits peut pourtant venir influencer la façon dont il veut se protéger vis-à-vis des mauvais résultats et par conséquent conditionner sa conception de la robustesse.

# The robust shortest path and the single-source shortest path problems : interval data

Martha Salazar-Neumann*

**Résumé**

Cet article traite de deux problèmes combinatoires robustes. Le premier concerne une version robuste du plus court chemin entre deux nøeuds préfixés 1 et m (trouver un plus court chemin robuste). Le second porte sur les chemins robustes entre un nøeud fixé et tous les autres. Dans ces deux problèmes, les coûts (ou longueurs) des arcs sont modélisés par des intervalles. On énonce des conditions suffisantes permettant de déterminer des arcs dits dits forts ou faibles appartenant ou non au plus court chemin robuste.

**Mots-clefs :** Plus court chemin robuste, modélisation de l'incertitude

**Abstract**

We consider two robust deviation problems; the robust deviation version of the problem to find a path of minimum weight connecting two specified nodes 1 and $m$, (*the robust deviation shortest path problem* ) and the robust version of the problem to find shortest paths from a fixed node 1 and all the nodes of the graph (*the robust deviation single-source shortest path problem* ). We consider both of them on finite directed graphs where arcs lengths are interval numbers. We give sufficient conditions for an arc $(k, r)$ to be always (for all realization of data) on an optimal solution (strictly strong arcs and T-strictly strong arcs resp.) and for an arc to be never on an optimal solution (non weak arcs and non T-weak arcs resp.). These entities can be used to preprocess a given graph with interval data prior to the solution of the robust problem. In [3] was shown that such preprocessing procedure is quite effective in reducing the time to compute the solution of the robust deviation path problem. Based on these results we present polynomial time recognition algorithms.

**Key words :** Minimax regret optimization; Robust shortest path problem; uncertainty modeling; arc problem

*SMG,Université Libre de Bruxelles, Boulevard du Triomphe - CP 210/01 B-1050 Bruxelles - Belgique. msalazar@ulb.ac.be

# 1 Introduction

The first problem that we consider is the robust deviation version of the problem to find a path of minimum weight connecting two specified nodes $1$ and $m$,*robust deviation shortest path problem* on finite directed graphs under arc length uncertainties. We model data uncertainty by treating the arc lengths as nonnegative interval numbers, where each arc length can take any value in its interval.

In Zielinski [8] this problem is shown to be NP-hard and to remain NP-hard even when a graph is restricted to be directed acyclic planar and regular of degree three.

In Averbakh et al. [1] authors prove that this problem is strictly NP-hard for acyclic directed graphs. Then in solving this problem, reducing the solution space becomes an important issue. In [3] authors show that for each optimal solution $p$ of the robust deviation shortest path problem there exists a scenario $s$ for which $p$ is an optimal solution of the deterministic problem corresponding to the scenario $s$.

They show also that knowing those arcs which are never on shortest paths we can preprocess a graph prior to solution of the robust path problem. They study the arc problems on acyclic directed graphs and present a procedure to check whether an arc incident at nodes $1$ or $m$ is weak or not. Then authors give a necessary condition for an arc to be non-weak give a procedure to determine if an intermediate arc is non-weak and propose a mixed integer programming approach with preprocessing for the problem. First, the mixed integer program is preprocessed by removing some arcs which will never be in an optimal path, and then it is solved by standard software. They show by computational results that preprocessing of graphs is an efficient method in solving the robust deviation path problem.

Montemani et al. [4], provide a branch and bound algorithm to solve the robust deviation path problem but they do not implement the preprocessing proposed in [3] because in practice it can be used only for acyclic layered graphs with small width.

Another way to reduce the solution space, is by detecting the arcs that are always on a shortest path from 1 to $m$. In Karasan et al. [3] authors study the robust deviation shortest path problem with interval data, unfortunately, no algorithm to detect the arcs that are always on the solution is given.

In another well-known robust combinatorial problem, the robust deviation spanning tree problem with interval data, Yaman et al. [7] present a mixed integer programming formulation with preprocessing for the problem. The preprocessing removes the non-weak edges and sets the variables corresponding to the strong edges equal to 1. They show by computational results that knowing non-weak and strong edges helps shorten significantly the computation of the robust deviation solution. In Salazar-Neumann [6] characterizations of strictly strong and non-weak edges for the robust deviation spanning tree problem with compact and convex uncertainty are given, and in the case of polyhedral uncertainty we give polynomial time recognition algorithms.

In this work we study the arc problem on a larger class of graphs, finite directed graphs. We give sufficient conditions for an arc to be always or never on a shortest path from 1 to $m$ (strictly strong arcs and non weak arcs respectively ). We present polynomial time algorithms to find some strictly strong arcs and some non weak arcs.

The second problem that we consider is the robust version of the problem to find shortest paths from a fixed node 1 to all the remaining nodes of $G$, the *single-source shortest path problem*. We give sufficient conditions for an arc $(k, r)$ to be always ( for all realization of data) on a shortest path from 1 to $r$ ( T-strictly strong arcs) and a characterization of the arcs that are never on a shortest path from 1 to another node of $G$ (non T-weak arcs).

Finally, we present polynomial time algorithms to find some T-strictly strong arcs and all the non T-weak arcs.

In [4] authors mention two important applications of this problem ; *"the transportation problems that are usually represented as a weighted digraph, where each arc is associated with a road and costs represent travel times. A shortest path problem has to be solved every time the quickest way to go from one place to another has to be calculated. A similar problem arises in telecomunications when a packet has to be sent from a source node to a destination node on a network. Also in this case, where the network is usually modeled as a weighted digraph and costs are associated with transmission delays, a shortest path problem is faced. In reality it is not easy to estimate arc costs exactly. To overcome this problem, the interval data model is an alternative".*

## 2 Notation and definitions

Most of the following definitions can be found in [2]. Let $G$ be a finite directed graph, we denote by $V(G)$ and $A(G)$, the set of nodes and arcs of $G$ respectively, we suppose that $\mid V(G) \mid = m$, $\mid A(G) \mid = n$. We consider two special nodes of $G$, the origin node 1 and the destination node $m$. We denote by $(i, j)$ the arc from node $i$ to node $j$. Let $A'$ be a nonempty subset of $A(G)$, the subgraph of $G$ with edge set $A(G) \setminus A'$ is written simply as $G - A'$ ; it is the subgraph obtained from $G$ by deleting the arcs in $A'$. If $A' = \{(i, j)\}$ we write $G - (i, j)$ instead of $G - \{(i, j)\}$.

Consider the following deterministic problem called shortest path problem on digraphs : Given $G$ and nonnegative length $l_{ij}$ associated with each arc $(i, j) \in A$, we want to find a shortest directed path connecting two specified nodes 1 and m, where the length of a path is the sum of the lengths of its arcs. An efficient $O(\mid V \mid^2)$ algorithm to solve this problem was given by Dijkstra. It finds not only a shortest (1,m)-path, but shortest paths from 1 to all nodes of $G$.

We will use some notations given in [3]. Let $S$ be the set of scenarios for the lengths of the arcs of $G$ and let $D$ be the set of our input data. We denote by $l_{ij}^s$ the nonnegative length of the arc $(i,j)$ in the scenario $s$ and we will assume that $D$ is a Cartesian product of intervals, that is to say, each $l_{ij}^s$ can take an arbitrary value in the interval $[\underline{l}_{ij}, \overline{l}_{ij}]$. We denote by $\underline{s}$ the scenario for which for all $(i,j) \in A$, $l_{ij}^{\underline{s}} = \underline{l}_{ij}$ and by $\overline{s}$ the scenario for which for all $(i,j) \in A$, $l_{ij}^{\overline{s}} = \overline{l}_{ij}$.

We denote by $P_1(G)$ the set of all the 1-paths of $G$, by $P_1^*(G,s)$ the set of all shortest 1-paths of $G$ under the scenario $s$ and by $P_{1j}(G)$ the set of all (1,j)-paths of $G$. For a path $p \in P_{1j}(G)$ we denote by $A(p)$ the set of arcs of $p$.

To evaluate a path $p \in P_1(G)$ under the scenario $s$ we use a function $f : P_1(G) \times D \rightarrow \mathbb{R} : (p, l^s) \rightarrow f(p, l^s)$ . In this work we assume that this function is the sum of the lengths of the arcs of $p$ under the scenario $s$.

$$f(p, l^s) = \sum_{(i,j) \in A(p)} l_{ij}^s.$$

By simplicity we will denote by $l_p^s$ the length of path $p$ in the scenario $s$. In the case where we want to find a shortest directed path connecting two specified nodes 1 and $m$. We define ;

$$f(p_s^*, l^s) = \min_{p \in P_{1m}(G)} f(p, l^s)$$

The first problem that we will consider in this work is the robust version of the shortest path from 1 to $m$ problem on digraphs under arc length uncertainties. The following definition can be found in the book [5] .

ROBUST DEVIATION : The robust deviation path $p_d$ is such that

$$\max_{s \in S}(f(p_d, l^s) - f(p_s^*, l^s)) = \min_{p \in P_{1m}(G)} \max_{s \in S}(f(p, l^s) - f(p_s^*, l^s))$$

where $p_s^* \in P_{1m}^*(G, s)$.

A worst deviation scenario for a path $p \in P_{1m}(G)$ is a scenario $s$ in which $f(p, l^s) - f(p_s^*, l^s)$, the difference between the cost of the (1,m)-path $p$ and the cost of a shortest path from 1 to $m$ in this scenario is maximum.

The following three definitions can be found in [3] ;

**Definition** A path is said to be a *weak path* if it is a shortest path from 1 to $m$ for at least one realization of arc lengths.

**Definition** An arc is a *weak arc* if it lies on some weak path.

**Definition** An arc is a *strong arc* if it lies on a shortest path from 1 to $m$ for all scenario.

Now we introduce the following definition ;

**Definition** An arc is a *strictly strong arc* if it lies on all shortest path from 1 to $m$ for all scenario.

The second problem that we will consider is the robust version of the problem to find shortest paths from a fixed node 1 to all the remaining nodes of $G$, the *robust single-source shortest path problem*. We consider this problem on digraphs under nonnegative arc length uncertainties. We call a feasible solution of the deterministic problem a *spanning 1-tree*, an optimal solution a *minimum spanning 1-tree* of $G$ and a shortest path from 1 to another node of $G$ a *shortest 1-path*.

We denote by $T^1(G)$ the set of all the spanning 1-trees of $G$, by $T_1^*(G, s)$ the set of all minimum spanning 1-trees of $G$ under the scenario $s$. For a spanning 1-tree $T \in T^1(G)$ we denote by $A(T)$ the set of arcs of $T$.

To evaluate a spanning 1-tree $T \in T^1(G)$ under the scenario $s$ we use a function $F : T^1(G) \times D \to \mathbb{R} : (T, l^s) \to F(T, l^s)$ . Where

$$F(T, l^s) = \sum_{(i,j)\in A(T)} l_{ij}^s.$$

By simplicity we will denote by $l_T^s$ the length of a 1-tree $T$ in the scenario $s$. We define ;

$$F(T_s^*, l^s) = \min_{T\in T^1(G)} F(T, l^s)$$

ROBUST DEVIATION : The robust deviation spanning 1-tree $T_d$ is such that

$$\max_{s\in S}(F(T_d, l^s) - F(T_s^*, l^s)) = \min_{T\in T^1(G)} \max_{s\in S}(F(T, l^s) - F(T_s^*, l^s))$$

where $T_s^* \in T_1^*(G, s)$.

A worst deviation scenario for a 1-tree $T \in T^1(G)$ is a scenario $s$ in which $F(T, l^s) - F(T_s^*, l^s)$, the difference between the cost of the spanning 1-tree $T$ and the cost of a minimum spanning 1-tree in this scenario is maximum.

In order to deal with the robust single-source shortest path problem we introduce the following concepts ;

**Definition** A path $p \in P_1(G)$ is said to be a *T-weak path* if $p$ is a shortest 1-path for at least one realization of arc lengths.

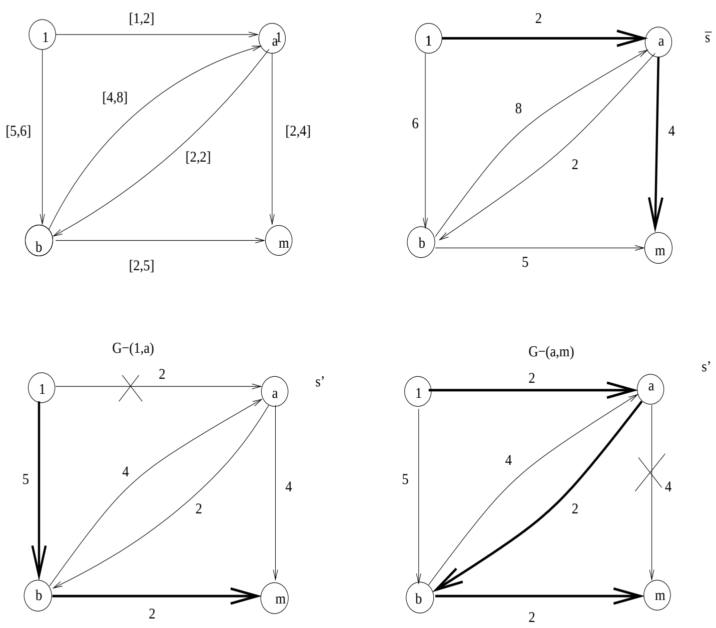



FIG. 1 – Strictly strong arcs algorithm

**Definition** An arc is a *T-weak arc* if it lies on some T-weak path.

**Definition** An arc $(i, j)$ is a *T-strictly strong arc* if it lies on all shortest path from 1 to $j$ for all scenario.

**Remark** We observe that an arc is a non T-weak arc, if and only if it never lies on a minimum spanning 1-tree and an arc is a T-strictly strong arc if and only if it lies on all minimum spanning 1-tree for all scenario.

# 3 The robust shortest path problem : strictly strong and non-weak arcs

The following proposition give a characterization of the strictly strong arcs.

**Proposition 1** Let $(k, r)$ be such that on the graph $G - (k, r)$ the node m is reachable from the node 1. For each $s \in S$ let $p^*_s \in P^*_{1m}(G, s)$. The arc $(k, r)$ is a strictly strong arc of $G$ if and only if for all $q \in P_{1m}(G - (k, r))$ and for all $s \in S$, $l^s_q > l^s_{p^*_s}$

**Proof :** Let $(k, r)$ be such that on the graph $G - (k, r)$ the node $m$ is reachable from the node 1. If $(k, r)$ is a strictly strong arc of $G$ and $q \in P_{1m}(G - (k, r))$, then $q$ is not a shortest path of $G$ from 1 to $m$ for each scenario $s$, and consequently for all $s \in S$,

$l_q^s > l_{p_s^*}^s$. Now, if for all $q \in P_{1m}(G-(k,r))$ and for all $s \in S$, $l_q^s > l_{p_s^*}^s$, then $q$ is never a shortest path of $G$ from 1 to $m$ for each scenario $s$, so for all $s \in S$ a shortest path of $G$ from 1 to $m$ under the scenario $s$ contains the arc $(k,r)$, and $(k,r)$ is then an strictly strong arc.

**Remark** Let $(k,r)$ be such that there exist $p \in P_{1m}(G)$ such that $(k,r) \in A(p)$ and on the graph $G-(k,r)$ the node $m$ is not reachable from the node 1, then the arc $(k,r)$ is a strictly strong arc.

We can observe that in the last characterization of the strictly strong arcs, the number of extreme scenarios can be exponential, then in the next theorem we will give a sufficient condition easy to test for an arc to be strictly strong. We will use the last proposition on the proof.

**Theorem 1** Let $G(V,A)$ be a finite directed graph, with origin node $1$ and destination node $m$. We suppose that each node $j \in V(G)$ is reachable from the node 1. Let $p$ be the shortest path from $1$ to $m$ found by Dijkstra algorithm under the scenario $\overline{s}$. Let $(k,r) \in A(p)$, and let $s'$ the scenario such that, $l_{ij}^{s'} = \overline{l}_{ij}$ if $(i,j) \in A(p)$, and $l_{ij}^{s'} = \underline{l}_{ij}$ if $(i,j) \notin A(p)$. If on the graph $G-(k,r)$ the node $m$ is reachable from the node 1, let $q' \in P_{1m}^*(G-(k,r),s')$ be a shortest path from 1 to $m$ under the scenario $s'$. If $l_{q'}^{s'} > l_p^{s'}$ then the arc $(k,r)$ is a strictly strong arc.

**Proof :** Let $p$ be the shortest path from $1$ to $m$ found by Dijkstra algorithm under the scenario $\overline{s}$, let $(k,r) \in A(p)$. Now consider the scenario $s'$ such that $l_{ij}^{s'} = \overline{l}_{ij}$ if $(i,j) \in A(p)$, and $l_{ij}^{s'} = \underline{l}_{ij}$ if $(i,j) \notin A(p)$. Let $q'$ a shortest path of $G-(k,r)$ from $1$ to $m$ under the scenario $s'$. If $l_{q'}^{s'} > l_p^{s'}$ then for all $q \in P_{1m}(G-(k,r))$ $l_q^{s'} \geq l_{q'}^{s'} > l_p^{s'}$ and then for all $q \in P_{1m}(G-(k,r))$

$$\sum_{(i,j)\in q\backslash p} \underline{l}_{ij} = \sum_{(i,j)\in q\backslash p} l_{ij}^{s'} > \sum_{(i,j)\in p\backslash q} l_{ij}^{s'} = \sum_{(i,j)\in p\backslash q} \overline{l}_{ij}$$

and then for all $s \in S$

$$\sum_{(i,j)\in q\backslash p} l_{ij}^{s} \geq \sum_{(i,j)\in q\backslash p} \underline{l}_{ij} > \sum_{(i,j)\in p\backslash q} \overline{l}_{ij} \geq \sum_{(i,j)\in p\backslash q} l_{ij}^{s}$$

this implies that for all $s \in S$ and for all $q \in P_{1m}(G-(k,r))$, we have $l_q^s > l_p^s$ and if $p_s^* \in P_{1m}^*(G,s)$, $l_q^s > l_p^s \geq l_{p_s^*}^s$. By proposition 1, $(k,r)$ is a strictly strong arc.

Theorem 1 allow us to derive a polynomial time algorithm to find some strictly strong arcs. An algorithm to find some non-weak arcs is presented in the following section.

**Algorithm to find some strictly strong arcs**

1. Apply Dijkstra algorithm to $G$ under the scenario $\overline{s}$ to obtain $p$, a shortest path from 1 to $m$.

2. Choose an arc $(k, r) \in A(p)$.
3. Construct the scenario $s'$ for which $l_{ij}^{s'} = \bar{l}_{ij}$ if $(i, j) \in A(p)$, and $l_{ij}^{s'} = \underline{l}_{ij}$ if $(i, j) \notin A(p)$.
4. Apply Dijkstra to $G - (k, r)$ under the scenario $s'$. If there exists a shortest path $q'$ from 1 to $m$, go to step 5. Otherwise $(k, r)$ is strictly strong.
5. Compute $l_{q'}^{s'}$ and $l_p^{s'}$.
6. If $l_{q'}^{s'} > l_p^{s'}$ then the arc $(k, r)$ is a strictly strong arc.
7. Go to step 2. Choosing another arc in $A(p)$ and continue.

**Example 1** To clarify the above procedure, we apply it on the first graph given in Figure 1. We set the lengths of all the arcs to their upper bounds and we find a shortest path $p$ from the node 1 to the node $m$, we represent such path with bold lines. We construct the scenario $s'$ setting the arcs $(1, a)$ and $(a, m)$ to their upper bounds and the lengths of the remaining arcs to their lower bounds. We delete the arc $(1, a) \in A(p)$ and we apply Dijkstra to $G - (1, a)$ under the scenario $s'$. As there exists a shortest path $q'$ from 1 to $m$, we compute $l_{q'}^{s'} = 7$ and $l_p^{s'} = 6$. As $l_{q'}^{s'} > l_p^{s'}$ then the arc $(1, a)$ is a strictly strong arc. Then we take the second arc $(a, m)$ of $p$, we delete the arc $(1, a) \in A(p)$ and we apply Dijkstra to $G - (a, m)$ under the scenario $s'$. As there exists a shortest path $q'$ from 1 to $m$, we compute $l_{q'}^{s'} = 6$ and $l_p^{s'} = 6$. As $l_{q'}^{s'} = l_p^{s'}$ we have not a conclusion for the arc $(a, m)$.

# 4 The robust single-source shortest path problem : T-strictly strong and non T-weak arcs

The following proposition give a characterization of the T-strictly strong arcs.

**Proposition 2** Let $(k, r) \in A(G)$ be such that on the graph $G - (k, r)$ the node r is reachable from the node 1. For each $s \in S$ let $p_s^* \in P_{1r}^*(G, s)$. The arc $(k, r)$ is a T-strictly strong arc of $G$ if and only if for all $q \in P_{1r}(G - (k, r))$ and for all $s \in S$, $l_q^s > l_{p_s^*}^s$.

**Proof :** Let $(k, r)$ be such that on the graph $G - (k, r)$ the node r is reachable from the node 1. If $(k, r)$ is a T-strictly strong arc of $G$ and $q \in P_{1r}(G - (k, r))$, then $q$ is not a shortest path of $G$ from 1 to $r$ for each scenario $s$, and then for all $s \in S$, $l_q^s > l_{p_s^*}^s$. Now if for all $q \in P_{1r}(G - (k, r))$ and for all $s \in S$, $l_q^s > l_{p_s^*}^s$ then $q$ is never a shortest path of $G$ from 1 to $r$ for each scenario $s$, so for all $s \in S$ a shortest path of $G$ from 1 to $r$ under the scenario $s$ contains the arc $(k, r)$, then $(k, r)$ is an T-strictly strong arc.

**Remark** Let $(k, r)$ be such that there exist $p \in P_{1r}(G)$ such that $(k, r) \in A(p)$ and on the graph $G - (k, r)$ the node $r$ is not reachable from the node 1, then the arc $(k, r)$ is a T-strictly strong arc.

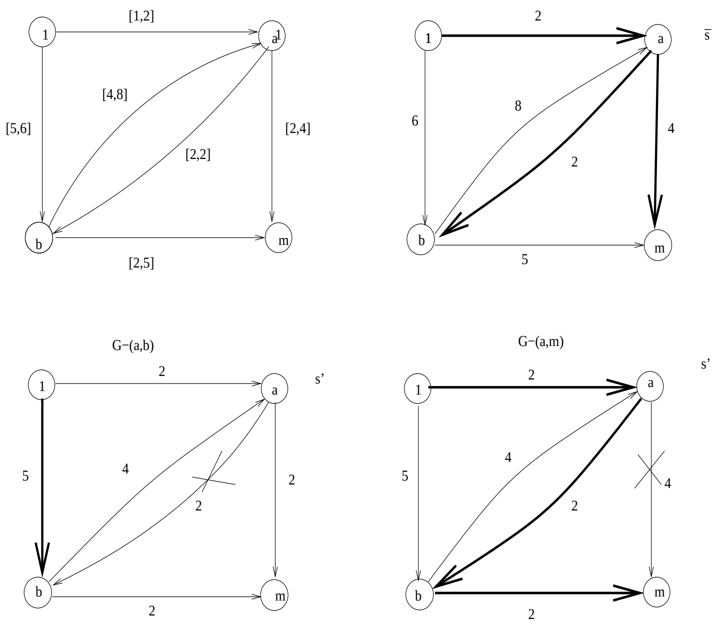



FIG. 2 – T-strictly strong arcs algorithm

On the last proposition, the number of extreme scenarios can be very large, and in that case we can apply the next theorem that give a sufficient condition easy to test for an arc to be T-strictly strong. The proof of next theorem uses the proposition 2.

**Theorem 2** Let $G(V, A)$ be a finite directed graph, with origin node $1$. Let $T$ the directed tree found by Dijkstra algorithm under the scenario $\overline{s}$, and let $(k, r) \in A(T)$. Let $p \in P_{1r}(G)$ such that $A(p) \subset A(T)$. Let $s'$ be the scenario for which $l^{s'}_{ij} = \overline{l}_{ij}$ if $(i, j) \in A(p)$, and $l^{s'}_{ij} = \underline{l}_{ij}$ if $(i, j) \notin A(p)$. If on the graph $G - (k, r)$ the node $r$ is reachable from the node 1, let $q' \in P^*_{1r}(G-(k, r), s')$ be a shortest path from 1 to $r$ under the scenario $s'$. If $l^{s'}_{q'} > l^{s'}_{p}$ then the arc $(k, r)$ is a T-strictly strong arc.

**Proof :** Let $T$ be the directed tree found by Dijkstra algorithm under the scenario $\overline{s}$, let $(k, r) \in A(T)$, and let $p \in P_{1r}(G)$ such that $A(p) \subset A(T)$. We observe that $p$ is a shortest path of $G$ from 1 to $r$ under the scenario $\overline{s}$.

Now consider the scenario $s'$ such that $l^{s'}_{ij} = \overline{l}_{ij}$ if $(i, j) \in A(p)$, and $l^{s'}_{ij} = \underline{l}_{ij}$ if $(i, j) \notin A(p)$. Let $q'$ a shortest path of $G - (k, r)$ from $1$ to $r$ under the scenario $s'$. If $l^{s'}_{q'} > l^{s'}_{p}$ then for all $q \in P_{1r}(G-(k, r))$ $l^{s'}_{q} \geq l^{s'}_{q'} > l^{s'}_{p}$ and then for all $q \in P_{1r}(G-(k, r))$

$$\sum_{(i,j)\in q\setminus p} \underline{l}_{ij} = \sum_{(i,j)\in q\setminus p} l^{s'}_{ij} > \sum_{(i,j)\in p\setminus q} l^{s'}_{ij} = \sum_{(i,j)\in p\setminus q} \overline{l}_{ij}$$

and then for all $s \in S$

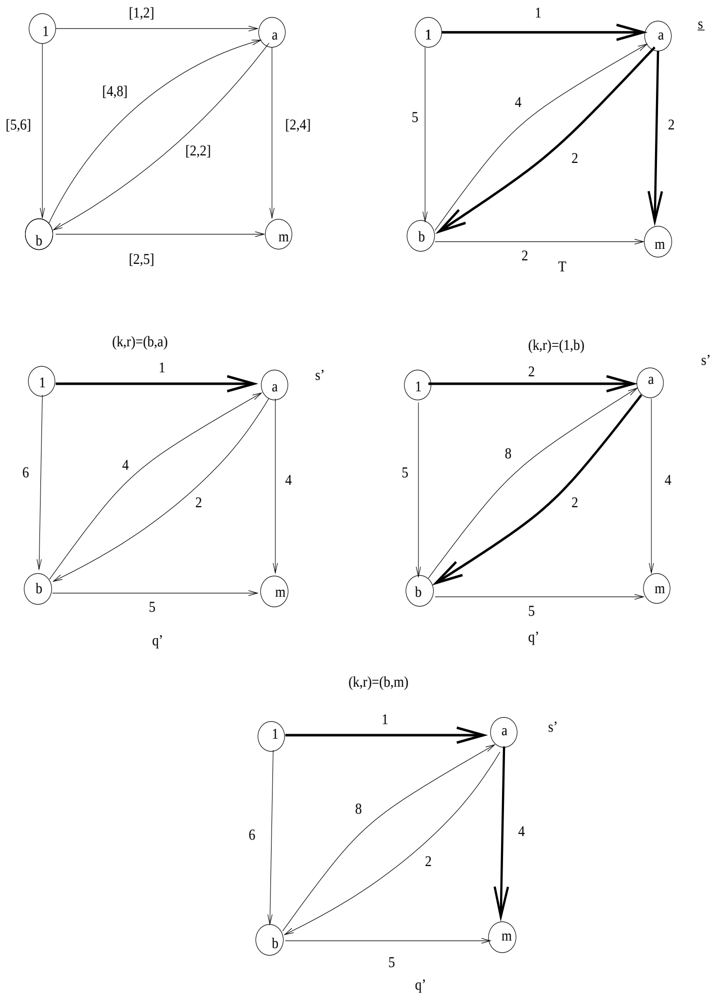



FIG. 3 – Non T-weak arcs algorithm

$$\sum_{(i,j)\in q\setminus p} l^s_{ij} \geq \sum_{(i,j)\in q\setminus p} \underline{l}_{ij} > \sum_{(i,j)\in p\setminus q} \bar{l}_{ij} \geq \sum_{(i,j)\in p\setminus q} l^s_{ij}$$

this implies that for all $s \in S$ and for all $q \in P_{1r}(G-(k,r))$, if $p^*_s \in P^*_{1r}(G,s)$, $l^s_q > l^s_p \geq l^s_{p^*_s}$ and then by proposition 2, $(k,r)$ is a T-strictly strong arc.

The following proposition give a characterization of the non T-weak arcs.

**Proposition 3** For each $s \in S$ let $p^*_s \in P^*_{1k}(G,s)$. The arc $(k,r)$ is a non T-weak arc of $G$ if and only if for each $s \in S$ there exist a path $q \in P_{1r}(G)$, such that $l^s_q < l^s_{p^*_s} + \underline{l}_{kr}$.

**Proof :** Let $(k,r)$ be a non T-weak arc of $G$, then for all $p^*_s \in P^*_{1k}(G,s)$, $p^*_s + (k,r)$ is never a shortest path of $G$ from 1 to $r$, so for all $s \in S$ there exists a path $q \in P_{1r}(G)$ such that $l^s_q < l^s_{p^*_s} + \underline{l}_{kr}$. Now if for all $s \in S$, exists a path $q \in P_{1r}(G)$ such that the inequality ; $l^s_q < l^s_{p^*_s} + \underline{l}_{kr}$ holds, then for all $p \in P_{1k}(G)$ $l^s_q < l^s_{p^*_s} + \underline{l}_{kr} \leq l^s_p + l^s_{kr}$ then all path of $G$ from 1 to $r$ that use the arc $(k,r)$ is never a shortest path from 1 to $r$ for the scenario $s \in S$, then the arc $(k,r)$ is a non T-weak arc.

The following theorem give another characterization easy to test for an arc to be a non T-weak arcs.

**Theorem 3** Let $G(V,A)$ be a finite directed graph, with origin node $1$. We suppose that each node $j \in V(G)$ is reachable from the node 1. Let $T$ the directed tree found by Dijkstra algorithm under the scenario $\underline{s}$ and let $(k,r) \in A \setminus A(T)$. Let $p \in P_{1k}(G)$ such that $A(p) \subset A(T)$. Let $s'$ be the scenario for which, $l^{s'}_{kr} = \underline{l}_{kr}$, $l^{s'}_{ij} = \underline{l}_{ij}$ if $(i,j) \in A(p)$, and $l^{s'}_{ij} = \bar{l}_{ij}$ if $(i,j) \notin A(p)$ and $(i,j) \neq (k,r)$. Let $q'$ be a shortest path of $G$ from $1$ to $r$ for the scenario $s'$. An arc $(k,r)$ is a non T-weak arc if and only if $l^{s'}_{q'} < l^{s'}_p + \underline{l}_{kr}$.

**Proof :** Let $T$ be the directed tree found by Dijkstra algorithm under the scenario $\underline{s}$, let $(k,r) \in A \setminus A(T)$, and let $p \in P_{1k}(G)$ such that $A(p) \subset A(T)$, then $p$ is a shortest path of $G$ from 1 to $k$ under the scenario $\underline{s}$. Consider the scenario $s'$ described before and let $q' \in P^*_{1r}(G,s')$. If $l^{s'}_{q'} < l^{s'}_p + \underline{l}_{kr}$

$$\sum_{(i,j)\in q'\setminus p} l^s_{ij} \leq \sum_{(i,j)\in q'\setminus p} \bar{l}_{ij} = \sum_{(i,j)\in q'\setminus p} l^{s'}_{ij} < \sum_{(i,j)\in p\setminus q'} l^{s'}_{ij} + \underline{l}_{kr}$$

then $l^s_{q'} < l^{s'}_p + \underline{l}_{kr}$ but $l^{s'}_p = l^{\underline{s}}_p \leq l^s_{p^*_s}$ then for all $s \in S$

$l^s_{q'} < l^{s'}_p + \underline{l}_{kr} \leq l^s_{p^*_s} + \underline{l}_{kr}$ and by proposition 8 that implies that $(k,r)$ is a not T-weak arc.

Now suppose that the arc $(k,r)$ is a non T-weak arc then by the proposition 8, for each $s \in S$ there exist a path $q \in P_{1r}(G)$, such that $l^s_q < l^s_{p^*_s} + \underline{l}_{kr}$. In particular for the scenario $s'$ there exists a path $q \in P_{1r}(G)$, such that $l^{s'}_q < l^{s'}_{p^*_{s'}} + \underline{l}_{kr}$. Consider the shortest path $q'$

from 1 to $r$ under the scenario $s'$ ($q' \in P^*_{1r}(G)$) then

$$l^{s'}_{q'} \leq l^{s'}_{q} < l^{s'}_{p^*_{s'}} + \underline{l}_{kr} \leq l^{s'}_{p} + \underline{l}_{kr}$$

and we have the result.

**Remark** All the non T-weak arcs are non-weak arcs.

The last two theorems allow us to derive polynomial time algorithms to find some T-strictly strong arcs all the non T-weak arcs and some non-weak arcs ;

**Algorithm to find some T-strictly strong arcs**

1. Apply Dijkstra algorithm to $G$ under the scenario $\overline{s}$ to obtain $T$.
2. Choose an arc $(k, r) \in A(T)$.
3. We consider the path $p \in P_{1r}(G)$ such that $A(p) \subset A(T)$.
4. Construct the scenario $s'$ for which $l^{s'}_{ij} = \overline{l}_{ij}$ if $(i, j) \in A(p)$, and $l^{s'}_{ij} = \underline{l}_{ij}$ if $(i, j) \notin A(p)$.
5. Apply Dijkstra to $G - (k, r)$ under the scenario $s'$. If there exists a shortest path $q'$ from 1 to $r$, go to step 5. Otherwise $(k, r)$ is T-strictly strong.
6. Compute $l^{s'}_{q'}$ and $l^{s'}_{p}$.
7. If $l^{s'}_{q'} > l^{s'}_{p}$ then the arc $(k, r)$ is a T-strictly strong arc.
8. Go to step 2. Choosing another arc in $A(T)$ and continue.

**Example 2** To clarify the above procedure, we apply it on the first graph given in Figure 2. We set the lengths of all the arcs to their upper bounds and we find shortest paths from the node 1 to all nodes of $G$. We represent such tree $T$ with bold lines. We choose the arc $(a, b)$ and we construct the scenario $s'$ setting the arcs $(1, a)$, and $(a, b)$ to their upper bounds and the lengths of the remaining arcs to their lower bounds. We delete the arc $(a, b) \in A(p)$ and we apply Dijkstra to $G - (a, b)$ under the scenario $s'$. As there exists a shortest path $q'$ from 1 to $b$, we compute $l^{s'}_{q'} = 5$ and $l^{s'}_{p} = 4$. As $l^{s'}_{q'} > l^{s'}_{p}$ then the arc $(a, b)$ is a T-strictly strong arc. Then we take a second arc $(a, m)$ of $T$, we construct the scenario $s'$ setting the arcs $(1, a)$, and $(a, m)$ to their upper bounds and the lengths of the remaining arcs to their lower bounds. We delete the arc $(a, m) \in A(T)$ and we apply Dijkstra to $G - (a, m)$ under the scenario $s'$. As there exists a shortest path $q'$ from 1 to $m$, we compute $l^{s'}_{q'} = 6$ and $l^{s'}_{p} = 6$. As $l^{s'}_{q'} = l^{s'}_{p}$ then we have not a conclusion for the arc $(a, m)$.

**Algorithm to find all the non T-weak arcs and some non weak arcs.**

1. Apply Dijkstra algorithm to $G$ under the scenario $\underline{s}$ to obtain $T$.
2. Choose an arc $(k, r) \in A \setminus A(T)$.

3. Consider the path $p \in P_{1k}(G)$ such that $A(p) \subset A(T)$.
4. Construct the scenario $s'$ for which $l_{kr}^{s'} = \underline{l}_{kr}$, $l_{ij}^{s'} = \underline{l}_{ij}$ if $(i,j) \in A(p)$, and $l_{ij}^{s'} = \bar{l}_{ij}$ if $(i,j) \notin A(p)$ and $(i,j) \neq (k,r)$.
5. Apply Dijkstra to $G$ under the scenario $s'$ to obtain $q'$ a shortest path from 1 to $r$.
6. Compute $l_{q'}^{s'}$ and $l_{p}^{s'}$.
7. If $l_{q'}^{s'} < l_{p}^{s'} + \underline{l}_{kr}$ then the arc $(k,r)$ is a non T-weak arc and then a non weak arc. Otherwise $(k,r)$ is a T-weak arc.

**Example 3** To clarify the above procedure, we apply it on the first graph given in Figure 3. We set the lengths of all the arcs to their lower bounds and we find shortest paths from the node 1 to all nodes of $G$. We represent such tree $T$ with bold lines. We choose the arc $(b,a) \notin A(T)$ and we consider a path $p \in P_{1b}(G)$ such that $A(p) \subset A(T)$. We construct the scenario $s'$ setting the arcs $(b,a)$, and $(1,a)$ to their lower bounds and the lengths of the remaining arcs to their upper bounds. We apply Dijkstra to $G$ under the scenario $s'$ to obtain a shortest path $q'$ from 1 to $a$, we compute $l_{q'}^{s'} = 1$, $l_{p}^{s'} = 3$ and $\underline{l}_{ba} = 4$. As $l_{q'}^{s'} < l_{p}^{s'} + \underline{l}_{ba}$ then the arc $(b,a)$ is a non T-weak arc. Then we take a second arc $(1,b) \notin A(T)$ and we consider a path $p \in P_{11}(G)$ such that $A(p) \subset A(T)$. We construct the scenario $s'$ setting the arc $(1,b)$, to their lower bound and the lengths of the remaining arcs to their upper bounds. We apply Dijkstra to $G$ under the scenario $s'$ to obtain a shortest path $q'$ from 1 to $b$, we compute $l_{q'}^{s'} = 4$, $l_{p}^{s'} = 0$ and $\underline{l}_{1b} = 5$. As $l_{q'}^{s'} < l_{p}^{s'} + \underline{l}_{1b}$ then the arc $(1,b)$ is a non T-weak arc. Finally we choose the arc $(b,m) \notin A(T)$ and we consider a path $p \in P_{1b}(G)$ such that $A(p) \subset A(T)$. We construct the scenario $s'$ setting the arcs $(b,a)$, and $(1,a)$ to their lower bounds and the lengths of the remaining arcs to their upper bounds. We apply Dijkstra to $G$ under the scenario $s'$ to obtain a shortest path $q'$ from 1 to $m$, we compute $l_{q'}^{s'} = 5$, $l_{p}^{s'} = 3$ and $\underline{l}_{bm} = 2$. As $l_{q'}^{s'} = l_{p}^{s'} + \underline{l}_{ba}$ then the arc $(b,m)$ is a T-weak arc.

**Conclusions and extensions :**

We have given sufficient conditions for an arc $(k,r)$ to be always on an optimal solution of the robust shortest path and for the single-source shortest path problems (strictly strong arcs and T-strictly strong arcs resp.) and for an arc to be never on an optimal solution (non weak arcs and non T-weak arcs resp.). Such conditions are based on the topology of the graph combined with the structure of the data set. Based on these results we have presented polynomial time recognition algorithms that can be used to preprocess a given graph with interval data prior to the solution of the robust problem. In [3] was shown that such type of preprocessing procedures are quite effective in reducing the time to compute the solution of the robust deviation path problem.

Future considerations in this area would involve the question : what are the cases when any shortest path algorithm would provide a robust deviation shortest path ? and the

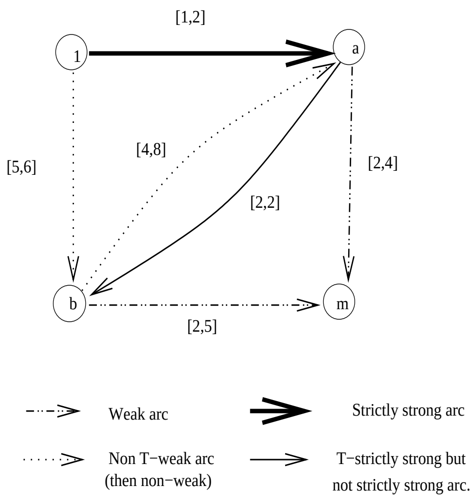



FIG. 4 – Classification of the arcs

investigation of the behavior of the number of arcs strong and non-weak on digraphs of big size.

# References


[1] I. Averbakh.and V. Lebedev. Interval data minmax regret network optimization problems. Discrete Applied Mathematics, 138(3): 289-301, (2004).

[2] J.A. Bondy and U.S.R. Murty. *Graph Theory with applications* American Elsevier, New York, and Macmillan, London.

[3] O.E. Karasan, M.C. Pinar, H. Yaman. The robust shortest path problem with interval data 2001. Available from http://www.optimization-online.org.

[4] R. Montemanni and L.M. Gambardella. A.V. Donati. A branch and bound algorithm for the robust shortest path problem with interval data. *Operations Research Letters*, 32:225-232, 2004.

[5] P. Kouvelis and G. Yu *Robust Discrete Optimization and its Applications.* Academic Pub. Boston,1997.

[6] M. Salazar-Neumann The robust minimum spanning tree problem: compact and convex uncertainty *Operations Research Letters* 35:17-22, 2007.

[7] H. Yaman, O.E.Karasan, M.C.Pinar. The robust spanning tree problem with interval data *Oper. Res. Lett.* 29, 31-40, 2001.

[8] P. Zielinski. The computational complexity of the relative robust shortest path problem with interval data. *Eur.J.Oper.Res.* 158, 570-576.

# Sensitivity and Robustness analyses in scheduling : the two machine with communication case

Eric Sanlaville*

**Résumé**

Nous considérons le problème de l'ordonnancement sous incertitudes. Deux analyses sont proposées pour le cas particulier de deux machines identiques avec incertitudes sur les délais de communication : une analyse de sensibilité des algorithmes optimaux en déterministe, et une analyse de robustesse. Toutes deux concernent le problème majeur de la recherche de solutions *robustes*. Les deux cas de petits et de grands délais de communication sont considérés séparément. Les études montrent le rôle central d'algorithmes processeur-ordonnés, c'est à dire des algorithmes qui bâtissent des affectations telles que toutes les communications se font d'une machine "émettrice" vers une machine "réceptrice". Ce type d'analyses peut être étendu à de nombreux autres problèmes d'ordonnancement.

**Mots-clefs :** Ordonnancement sous Incertitudes, Délais de Communication, Sensibilité et Robustesse

**Abstract**

The problem of scheduling under uncertainties is considered. In the special case of two identical machines with uncertainties on the communication delays, two analyses are presented: sensitivity analysis for deterministic optimal algorithms, and robustness analysis. Both deal with the major issue of finding *robust* solutions. The two cases of small and large communication delays are separately dealt with. Both studies show the central role of processor-ordered algorithms, that is algorithms building assignments such that all communications go from one "sending" machine to one "receiving" machine. This kind of analyses can be extended to many other scheduling problems.

**Key words :** Scheduling under Uncertainties, Communication Delays, Sensitivity and Robustness

*LIMOS, Université Blaise pascal, Campus des Cézeaux, 63177 Aubière Cedex – France eric.sanlaville@isima.fr

# 1 Preliminaries

## 1.1 Goal of the paper

Building robust solutions is a paramount issue in modern applications of scheduling. Often, the goal is to use robust methods, in the sense that the resulting schedules will have a good behaviour whatever the actual conditions of execution of the schedule may be. In some cases, the solution schedules can, with good results, be computed off-line by a so-called *proactive* algorithm: one solution is computed by an algorithm that takes into account the uncertainties on some data. This paper considers a very specific problem for which a proactive approach seems sufficient. This is the 2 machine with communication delays problem, with tree like precedence constraints and makespan criterion. The communication delays are not deterministic. The goal of the paper is to present an exhaustive study (based on already published as well as new results) of the robustness for this problem in its proactive version: what are the most robust proactive agorithms ? The problem is sufficiently simple, so that it is possible to get conclusive results to this question. The feeling of the author is that similar results should be obtained for other, maybe more complicated, scheduling problems.

## 1.2 The model

Let us consider the following model: A set of $n$ tasks bound by precedence constraints is to be executed on a set of identical parallel machines. The tasks have arbitrary durations. Moreover, if two tasks bound by a precedence constraint are executed on different machines, there is a so-called communication delay between the end of the first and the beginning of the second task. Classical restrictions in scheduling apply: a task cannot be executed simultaneously on different machines, one machine executes one task at a time. Moreover, the overlapping of communications by executions is allowed: A task may begin on one machine while this machine is communicating. The optimization criterion is the makespan. The study focuses on two machines and a tree structure for the precedence graph. This problem is central as the deterministic version is polynomial under some additional hypotheses on durations. However, the above theoretical model is quite general. In a workshop the communication delay models the transportation time of the part from one workstation to another (sometimes called minimum time lag). In parallel computing it models the time to send data from one processor to another (the result of one computation being used as the entry for another computation done on another machine).

## 1.3 Some additional definitions and notations

In most models, the communication delay depends on the initial task $T_i$ and the final task $T_j$, and not on the machines executing them. It is denoted by $c_{ij}$. When execution durations (denoted by $p_j$, $j = 1 \ldots, n$) are unitary the problem is called a UET problem (Unit Execution Time). When communication delays are unitary too, it is called UECT. If the maximum (respectively minimum) communication delay is smaller (resp. larger) than or equal to the smallest (resp. largest) execution duration, the problem is called SCT or coarse granularity problem (resp. LCT or small granularity problem). We shall also use notations UET-SCT and UET-LCT. The basic problem studied in the paper is written in classical scheduling notation $P2 \mid tree, uect \mid C_{max}$.
A schedule $S$ is defined by, for each machine $M_i$:

- a set of tasks assigned to $M_i$ .
- an order (sequence) on these tasks on $M_i$
- the beginning times of each task.

$\pi_j(S)$ is the number of the machine assigned to $T_j$ in the schedule $S$. If $T_i \prec T_j$, and $\pi_i(S) \neq \pi_j(S)$, the communication between $T_i$ and $T_j$ is said *effective* for $S$.
$\omega_c(S)$ denotes the makespan of $S$ for the vector $c$ of communication delays, and $\omega_c^*$ is the optimal makespan for this vector. In the special case where $c_{ij} = k$ for all precedences $T_i \prec T_j$ (that is, $c$ is a constant vector), notations $\omega_k(S)$ and $\omega_k^*$ are used.

## 1.4 Main results in deterministic case

The 2 machine, UECT case deserves specific studies. It is a frontier problem for complexity, as $P2 \mid prec, uect \mid C_{max}$ is still open (as $P3 \mid prec, uet \mid C_{max}$ itself). List methods are extended in different ways to take communications into account. We shall here consider the most natural and most used extension (see Hwang et al. [10], Hanen and Munier [8]). As usual, some task $T_k$ is said *ready* at time $t$ if its execution may begin at $t$ on at least one machine. But because of the communications, there are two kinds of ready tasks. If each predecessor $T_j$ of $T_k$ is finished before $t - c_{jk}$, $T_k$ might begin at $t$ on any machine and is called *free*. Otherwise, one predecessor of $T_k$, say $T_{\bar{\jmath}}$ finished after $t - c_{\bar{\jmath}k}$, and $T_k$ cannot be executed on another machine than $\pi_{\bar{\jmath}}$. It is ready, but *bound* to this machine. The algorithm assigns, whenever machine becomes available, the task of largest priority among free tasks and tasks bound to this machine. Note that there might be several ready tasks bound to the same machine, and at most one of these tasks can be executed at $t$. Hence a task may be executed before another one of larger priority, if the second one is bound to an unavailable machine.

This extension is called ETF (Earliest Task First) in [10]. Algorithm ETF without priority has an approximation ratio of 2 in the UECT arbitrary graph case. The most frequently used priority is based upon the Critical Path: tasks are sorted according to the longuest path between them and a task without successor. This variant is denoted by ETF/CP. Accordingly, the algorithm ETF/CG ( Coffman and Graham priority, strictly included in CP) has an approximation ratio of $\min(3/2, 4/3 + 3/w^*)$ which is reached, see [8]. Problem $P2 \mid tree, uect \mid C_{max}$ is polynomial. Surprisingly, 4 different algorithms were proposed to solve it, due to Picouleau in 92 [17], Veldhorst in 93 [19], Lawler in 93 [11], and Guinand and Trystram in 95 published in [6]. The most well known one is that of Lawler, presented for out-trees. It can be extended immediately to $m$ machines, still keeping good performances: Guinand, Rapine and Trystram [5] show an absolute approximation bound of $(m-1)/2$ . Möhring and Schäffter [15] studied this algorithm and extended it to communication delays either unitary or null. This extension preserves the previous results: optimality for 2 machines, and the bound for $m$ machines. An interesting point from their work is that Lawler algorithm is considered as a list schedule of ETF kind (as such, it can be used for arbitrary execution times and communication delays). The priority of one task $T_j$ is given by a height value $h(j)$ defined as the duration of an optimal schedule of the tree rooted in $T_j$ on an unbounded number of machines. The computation of $h$ is very simple, and follows exactly the method of [2]. Hence we shall from now on denote this algorithm by ETF/LMS.

Another point is the fact that ETF/CP schedules are not optimal in general for out-trees, and not also for in-trees, even if it is still open to know if such schedules are dominant. Indeed, direct application of LMS rule on in-trees is equivalent to CP rule, but an ETF/LMS schedule read in reverse order, with tasks executed as early as possible( hence optimal for out-trees), does not respect CP priority.

Guinand and Trystram in [6] proposed an algorithm named CBOS (Clusters Based on Subtrees) for in-trees. Starting from the root assigned to the first machine, it assigns on the second machine entire sub-trees whose cardinality respects some maximality condition. This condition implies that at most $(n-2)/2$ tasks are thus assigned to the second machine. The tasks on the second machine are executed subtree by subtree, following their root level. The other tasks are executed by the first machine as soon as they are ready.

The algorithm may be used with non unitary delays, with minor changes. In the constant case ($c_{ij} = k$) CBOS admits some performance guarantees : an absolute bound of 1 if $k = 2$, of $k/2 + 1$ for arbitrary $k$. This problem is in fact pseudo-polynomial (there exists a dynamic programming algorithm of complexity $O(n^{k+1})$). If the delays are unitary but the execution times are in $\{1, \ldots, l\}$, the absolute bound is $\lceil l/2 \rceil$ .

## 2 Taking uncertainties into account: motivations, approaches, notations

Deterministic models are used only in a first phase in the management of real applications. Indeed, for applications in Operations Research in general (for instance, localisation, network design, planning) and particularly in scheduling, uncertainty on the data is more a rule than an exception. This study focuses on uncertain communication delays. Indeed, whatever application is considered (production, parallel computing) these delays ( transportation times in a workshop, sending times in an interconnection network) are either uncertain by nature, or extremely difficult to compute precisely. Of course, the execution times, the machine or human operator availabilities, the ready times of the tasks, … are also subject to uncertainty, and examples may be found in recent literature, see [1].

Several approaches are possible for taking uncertainties into account, namely the *proactive, reactive, and proactive-reactive* approaches. In proactive approach a schedule is computed with the available data before execution phase (these data encompass the hypotheses on uncertainties, perturbations, that are done on these data). During execution one tries to stay as close as possible to this schedule, sometimes called baseline schedule, see [9]. The term off-line schedule may also be used, especially in parallel computing. In the reactive approach, no schedule is built before execution. During execution, the schedule is built step by step, with the help of simple rules often called decision rules. The schedule is called on-line. List algorithms can be used in such an approach. The proactive-reactive approach naturally starts from a baseline schedule, but modifies it online so that it better fits the actual conditions (actual data values). The frontier between proactive and proactive-reactive approaches is somewhat fuzzy: practically, a minimal flexibility must be attached to the baseline schedule. Usually, temporal flexibility at least is allowed: the starting and finishing times of the tasks may move, but the execution order of the tasks on each machine stay fixed. If some flexibility on the task order is allowed (a task exchange is possible even if the first task in the baseline schedule can still be executed in the same machine) or even on the task assignment, then we are in the proactive-reactive framework.

The choice between the three paradigms depends of course on the magnitude of the uncertainties, but also on the freedom available at execution time (exactly the flexibility above). Remark that it is not always advisable to use the full extent of this given flexibility. If the performances of some baseline schedule remain good whatever the real data, it is better to keep this schedule than to try to improve it and loose some stability. Such schedule is called *robust*. This naturally leads to ways to quantify this appreciation.

## 2.1 Robustness measures

Uncertainty may be represented either by stochastic models, either by interval models. In the first case, data are associated to random variables. In the second case, to intervals or to discrete sets of values (a *scenario* is then obtained by fixing a value to each quantum of data). This paper studies the second case. There is no need for hypotheses on data distributions, but it follows that the analyses are worst case analyses, and not mean analyses. With deterministic hypotheses, it suffices to choose one criterion $Z$, like the makespan, to measure a schedule performance. An algorithm is then evaluated according to this criterion value for all instances of the problem. When uncertainties are considered, an instance of the problem contains all possible scenarios. The worst case robustness of a schedule is obtained, for one given instance, by considering its characteristics for each scenario. Quite often, the characteristic used is simply the value of $Z$ in each scenario.

The robustness of an algorithm, or more generally of the whole building process of a final schedule, is evaluated according to the robustness of the schedules it builds for all considered instances. These schedules are the solutions obtained from the algorithm for each problem-with-uncertainty instance. There are several ways to make this evaluation, these are called *robustness metrics* in the following.

The first kind of metric is *stability*. The goal is to measure the variations between the schedules obtained for the different scenarios. A stable algorithm builds schedules which differ as little as possible. Maximizing stability does not take into account the value of $Z$. Hence other metrics use these values. *Threshold metrics* compute the distance between $Z$ and some given value $\overline{Z}$ that ideally should not be exceeded. Looking for algorithms that minimize this metric is a cautious approach, but it might be unappropriate to propose solutions that can be far from optimal for some scenarios. In this paper, *deviation metrics* are used for worst case robustness measure. For each scenario, the deviation between $Z$, the obtained performance and $Z^*$, optimal performance for the scenario, is computed. The worst case robustness of a schedule is obtained as this largest deviation. The deviation is relative if it is computed as $Z/Z^*$ (one may also encounter "competitive ratio", edspecially in on-line literature), absolute if it is $Z - Z^*$ (or "regret"). The robustness of an algorithm is of course obtained as the largest deviation of the schedules it produces.

## 2.2 Sensitivity analysis or Robustness analysis?

After the choice of the robustness measure is done, there are still several possibilities to use it. The first is the sensitivity analysis. It is associated with one particular algorithm, and in a broad sense consists in computing its robustness. Examples are numerous in recent papers, see [7, 4, 16, 18]. Note that the above definition is more general than what is usually considered as sensitivity, particularly in linear programming: the domain of one fixed parameter, around some reference value, for which some solution remains optimal.

In real problems there is not always such solution, or the disturbances can be much higher than the size of the domain, or concern several parameters simultaneously.

The second is simply called robustness analysis, or robust optimization as in fact it is an optimization problem: find the algorithm whose robustness is optimal for a given metric. If sensitivity analysis is always possible with reasonable hypotheses, robustness analysis can be a very difficult optimization problem. Examples in the literature are few. Hall and Posner [7] deal with $1 \mid\mid \sum w_j U_j$, Daniels and Kouvelis [3], Lebedev and Averbakh [12] consider $1 \mid\mid \sum C_j$, with uncertainties on the execution times. The two last papers use the worst case absolute deviation ("minmax regret"). [3] shows the problem is NP-complete with just two scenarios ; [12] looks at the interval model: the problem is still NP-complete unless all intervals are identically centered, and the number of tasks is pair!
It is well known that the list algorithm SPT is optimal in the deterministic case of $1 \mid\mid \sum C_j$. Unfortunately there is no sensitivity analysis of SPT for the absolute deviation. In [16], such an analysis is achieved for the relative deviation, for one or $m$ machines. In the centred case (all tasks have same mean), every order is SPT. The algorithm of [12] that optimizes robustness consists in putting in the sequence middle the tasks of largest interval. The general robustness problem with minmax regret being NP-complete, SPT does not achieve optimal robustness. But the above results hint that it might just be the best possible polynomial method.

Mahjoub [14] studies $1 \mid\mid \sum U_j$, with uncertainties on the machine availability: the machine might be unavailable at time $0$. This problem is interesting for two reasons. In its deterministic version, it is polynomial and is solved in $O(n \log n)$ by a slight modification of EDD rule ( Earliest Due Date first), but the optimal solution is extremely sensitive to perturbations. Hence it is important to look for more robust solutions. The absolute deviation measure is used. The robustness optimization problem remain polynomial with reasonable hypotheses, even if the unavailability time is in some real interval $[0, S]$, or in a finite set $\Omega$ of scenarios.

One of the goals of the present paper is to show the complementarity of these two analyses. The results of sensitivity analysis for a given algorithm will be usefully compared with the properties of an optimal robust schedule, even if this schedule is impossible to use in practice. The problem studied in the rest of the paper is simple enough, so that optimizing robustness remains an attainable objective. Last, it is of course possible to start from some given algorithm and to increase its robustness (see [7]) a paradigm the automaticians call robustification (in a reactive framework).

## 2.3 Proactive approach in the 2 machine case

As mentionned earlier, we focus on proactive approaches for the problem under study, that is $P2 \mid tree, uect \mid C_{max}$, with uncertain communication delays. The estimated delays are equal to $1$, all actual delays $c_{ij}$ are reals in the same time interval $[\underline{c}, \overline{c}]$, with $\underline{c} \leq 1 \leq \overline{c}$. In this context, a proactive approach computes a first schedule, that can be in a small extent modified on-line. Here is the exact complete process.

A complete schedule (assignment, sequences and dates) is determined by an off-line algorithm using estimated delays. The final schedule admits the following properties:

- The assignment is not modified,
- the execution order of the tasks on each machine is not modified,
- The final starting times of the tasks are obtained by executing the tasks as early as possible.

**Definition 1** A relative sensitivity bound *of an algorithm $\mathcal{A}$ upper-bounds the maximum ratio* $\dfrac{\omega_c(S)}{\omega_c^*}$ *over all considered communication delays, $S$ being the schedule built by $\mathcal{A}$.*

**Definition 2** An absolute sensitivity bound, *when it exists, of an algorithm $\mathcal{A}$ upper-bounds the maximum difference $\omega_c(S) - \omega_c^*$ over all considered communication delays, $S$ being the schedule built by $\mathcal{A}$.*

Both bounds depend on $\overline{c}$ and $\underline{c}$, they are upper bounds of worst case relative and absolute deviations when effective communication delays are in $[\underline{c}, \overline{c}]$. The relative sensitivity bound, or competitive ratio, is the equivalent of the approximate ratio in robustness context (see section 1.4).

# 3 General results

Some of the presented results are proved in [4]. In what follows, the task graph has an in-tree structure unless otherwise mentioned.

**Definition 3** *A* processor-ordered assignment *(*PO*-assignment) is such that the effective communications are all in the same direction between the two machines. A* PO*-schedule is such that its assignment is* PO*.*

Remark: Let $S$ be a PO-schedule. The schedule obtained from $S$ on the reversed graph (by reversing the arcs) is still PO. Without loss of generality, each schedule executes the root of the tree on machine $M_r$ (for receiving machine), the second machine is denoted by $M_s$ (for sending machine). Hence an assignment is PO if and only if, the execution of a task by $M_r$ entails that all its successors are also executed by $M_r$.

Note that PO-schedules can be considered even for zero communication delays, this and the two results below will be used later in the paper.

**Theorem 1 (in [4], theorem 1)** *For each instance of $P2 \mid tree, uet \mid C_{max}$, there is some optimal* PO*-schedule.*

Such a schedule is obtained from Hu's algorithm (list algorithm, level by level priority equivalent to CP priority, optimal in the UET case), by a sequence of task exchanges.

**Theorem 2 (in [4], theorem 2)** *For each instance of $P2 \mid tree, uet \mid C_{max}$, and for each communication delay vector $c$, the difference between the optimal makespans with and without communication delays is bounded by $\overline{c}$ : $\omega_c^* - \omega_0^* \leq \overline{c}$.*

Indeed, the maximum makespan increase of an optimal PO-schedule for null delays is clearly of $\overline{c}$. The result then reads directly from theorem 1. This is illustrated by figure 1: the schedule's end is delayed at most by $\overline{c} = 1.5$.

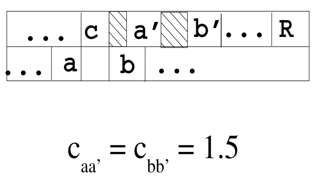



Figure 1: Illustration of theorem 2

# 4 Studies for small communication delays

In this section, optimal algorithms for $P2 \mid tree, uet - sct \mid C_{max}$ in the deterministic case are considered; their performance is evaluated with regard to $\underline{c}$ ( $\overline{c} = 1$).

## 4.1 Sensitivity analysis

Examples can be built to show that PO-schedules are not dominant for SCT problems, as they are for null delays ( theorem 1). Still, any optimal algorithm in the UECT case keeps interesting properties in the SCT case. Suppose $\underline{c} > 0$. Then the following is true:

- if $\omega_0^* = \omega_c^*$, then every optimal schedule is without idleness on $M_r$,
- if $\omega_0^* < \omega_c^*$, and if there is some optimal schedule such that the idleness duration on $M_r$ is at least equal to $\underline{c}$, then $\omega_0^* + \underline{c} \leq \omega_c^*$,
- if $\omega_c^* - \omega_0^* < \underline{c}$ then $\omega_0^* = \omega_c^* = \omega_1^*$.

The theorem follows from these properties.

**Theorem 3 (in [4], theorem 3)** *an optimal algorithm for* UECT *trees admits an absolute sensitivity bound in the* SCT *case of at most* $1 - \underline{c}$*:*

$$\omega_c - \omega_c^* \leq 1 - \underline{c}$$

Hence, this result is verified for the $4$ previously mentioned algorithms. For each of these, the bound is reached.

## 4.2 Robustness analysis: minimization of the absolute deviation.

Let $R_{\underline{c}}(S) = \max_{c \in [\underline{c},1]} \omega_c(S) - \omega_c^*$. We shall say that a schedule is robust for the absolute deviation measure *and for a fixed value of* $\underline{c}$, if it minimizes $R_{\underline{c}}$. We shall see that in the UET-SCT case, the robustness is in fact independent of $\underline{c}$. The CBoS algorithm tries to minimize the number of communications. Furthermore, it sometimes assigns too few tasks to $M_s$ to keep optimality for small communication delays. Let us consider the following example: two tasks $1$ and $2$ precede the root. CBoS assigns the three tasks to $M_r$. This schedule is optimal if the communication delays are unitary, but not when they are less than one (the makespan difference is then at most $1 - \underline{c}$ ). The schedule executing two tasks on $M_r$ is optimal as soon as the communication vector $c$ is constant. But there is a communication vector for which both schedules behave as badly. In fact they are both non optimal for all possible vectors, but they are robust in the above sense, as the next theorem will show.

Remark : from theorem 2, $\omega_1^*$ is equal either to $\omega_0^*$ or to $\omega_0^* + 1$. In the first case any optimal schedule for unitary delays is optimal for any SCT communication vector. In the second case $\omega_c^* = \omega_1^* - 1 + \underline{c}$.

**Theorem 4** *suppose* $\omega_1^* = \omega_0^* + 1$. *Then for any* $S$, $R_{\underline{c}}(S) \geq 1 - \underline{c}$.

**Proof.** If $S$ is not optimal considering unit delays, for integrity reasons $\omega_1(S) \geq \omega_1^* + 1$ hence $R_{\underline{c}}(S) \geq 1$. So let us suppose that $S$ is optimal for $c = 1$. If $M_r$ is never idle in $S$ for unit delays, the makespan of $S$ is the same whatever the delays are, hence $\omega_c(S) = \omega_c^* + 1 - \underline{c}$ and $R_{\underline{c}}(S) \geq 1 - \underline{c}$.
Hence suppose there are some idleness slots on $M_r$ For $S$ when $c = 1$. Let $\theta$ be the starting time of the latest idle slot, $T_i$ and $T_j$ such that the communication between $T_i$ on $M_s$ and $T_j$ on $M_r$ entails this idleness. Let $c$ be such that the delay is equal to 1 for any communication that begins before $\theta$, and also for $(i, j)$, and equal to $\underline{c}$ after. As there is no idleness after $\theta$, $\omega_c(S) = \omega_1(S) = \omega_1^*$. Let $S'$ be the schedule obtained from $S$ by exchanging the assignment of all tasks between 0 and $\theta$. In $S'$, $T_j$ can be shifted forward of $1 - \underline{c}$, as $(i, j)$ is not an effective communication any more. All tasks assigned to $M_r$ after $T_j$ by $S$ are at most at 1 time unit apart from a possible predecessor on $M_s$. Hence they can be shifted forward of the same duration as $T_j$. So $\omega_c(S) = \omega_c(S') + 1 - \underline{c}$. ∎

From the remark above, when $\omega_0^* = \omega_1^* = \omega_c^*$ for any vector $c$, any optimal schedule for unit delays (or simply for non zero delays), optimizes the robustness. Conversely, if $\omega_0^* + 1 = \omega_1^*$, the robust schedules are those which are optimal for unit delays, from theorem 3. Note that a schedule, like those built by CBOS, is robust, as its makespan is constant for any SCT-communication vector. Let us consider some PO-schedule optimal for zero delays (it is sure such schedule exists, see theorem 1). It is optimal for constant delays, hence for $c = 1$ (unitary delays), if $\omega_0^* + 1 = \omega_1^*$, and optimizes the robustness for this case. Unfortunately if $\omega_0^* = \omega_1^*$, it is not always optimal for non zero delays.

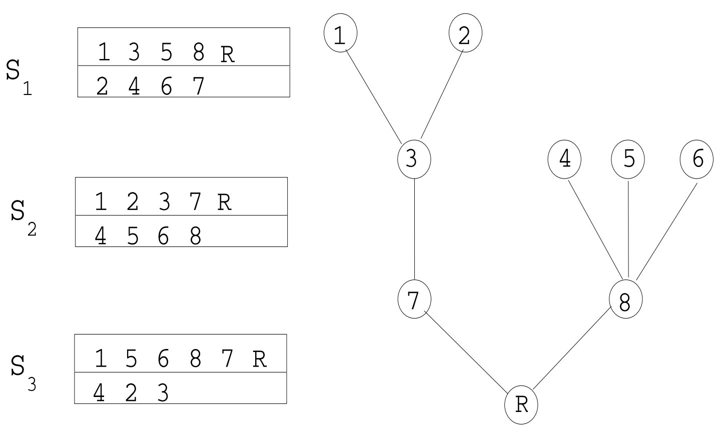



Figure 2: Three different schedules with $R_c = 1 - \underline{c}$, for any value of $\underline{c}$

To summarize, any algorithm that builds optimal schedules in the UECT case is robust. Furthermore if $\omega_0^* = \omega_1^*$, such an algorithm (as CBOS or ETF/LMS) builds optimal schedules whatever the delays are. If $\omega_0^* + 1 = \omega_1^*$, no algorithm can exist with smaller $R$ than $1-\underline{c}$. Example of figure 2 illustrates this: suppose $\underline{c} = 0.25$, and consider the three schedules of the figure (represented without communication delays). Schedule $S_3$, obtained by Lawler's algorithm, has a makespan of $6$ for any communication vector. If $c = 0.25$ for any communication delay, $S_2$ and $S_3$ are optimal with makespan $5.25$. But if $c_{8R} = 1$ (respectively $c_{68} = 1$) and the other delays keep the same value of $0.25$, $\omega(S_1) = 5.25$ and $\omega(S_2) = 6$ (respectively $\omega(S_2) = 5.25$ and $\omega(S_1) = 6$). Hence the three schedules have a robustness of $0.75 = 1 - \underline{c}$. It is easy to verify that the result is verified for the three schedules for any value of $\underline{c}$.
Note however that it is always possible to build a PO-schedule, both robust and optimal for constant delays, as $S_2$ (which could not however be obtained by CBOS) in the example. Hence PO-schedules are dominant for the optimization of robustness, in particular CBOS is robust in any case, but also for deterministic makespan minimization with constant delays.

# 5 Study for large communication delays

For UET-LCT problems, a relative bound can be obtained:

**Theorem 5 (in [4], theorem 4)** *Optimal schedules for* UECT *trees admit a relative sensitivity bound in the* LCT *case of at most* $\dfrac{\overline{c}+1}{2}$.

For PO-schedules, the following result holds:

**Theorem 6 (in [4], theorem 5)** *Let us consider a problem such that* $\overline{c} \geq 1 \geq \underline{c}$. *Optimal* PO-*schedules for* UECT *intrees admit an absolute sensitivity bound of at most* $\overline{c} - \underline{c}$.

The three first algorithms already mentioned do not build PO-schedules. And indeed it is possible for each to build instances for which the bound is asymptotically reached (it follows that no absolute bound exists for these). Conversely, CBOS of course builds PO-schedules. From CBOS optimality, it follows that for any instance of $P2 \mid tree, uect \mid C_{max}$, there is always an optimal PO-schedule. The above results concerning PO-schedules imply that CBOS admits an absolute sensitivity bounds equal to $\overline{c} - \underline{c}$. Is it a robust algorithm ? Unfortunately, the answer is no.

Indeed, in the UET-LCT case, examples may be built where CBOS is dominated by another schedule each time the delays are in $[1, \overline{c}]$. This is shown in figure 3. In this example, no PO-schedule is optimal for a set of communication vectors. Indeed the first schedule $S_1$ is such that the time interval between two tasks with effective communication is always two. This schedule is optimal if $\overline{c} \leq 2$ (it is *stable* for $\overline{c} \in [1, 2]$, which is in a way an even stronger result). But this delay is at most $1$ for at least one communication for any optimal (UECT case) PO-schedule (the second schedule is obtained by CBoS). This implies that for PO-schedules the bound of $\overline{c} - 1$ is reached if $\overline{c} \leq 2$. For larger values of $\overline{c}$, this is still true, but the communication vector depends on the PO-schedule. For $S_2$, it suffices to compare it with $S_1$, where tasks number $1$ and $4$ are exchanged, and to set $c_{45}$ to $\overline{c}$, and the other delays to $1$.

Remark : In the important case of equal actual delays, *processor-ordered* schedules have a makespan of at most $\omega_0^* + c$. Any other schedule has a makespan larger than or equal to $3 + 2 \times c$ (if any path of the graph contains at most two effective communications). The schedule that assigns all tasks to $M_r$ (*trivial schedule*), with a constant makespan on $n$, is a particular PO-schedule. This implies that PO-schedules are dominant when minimizing the makespan, for sufficiently large values of $c$.

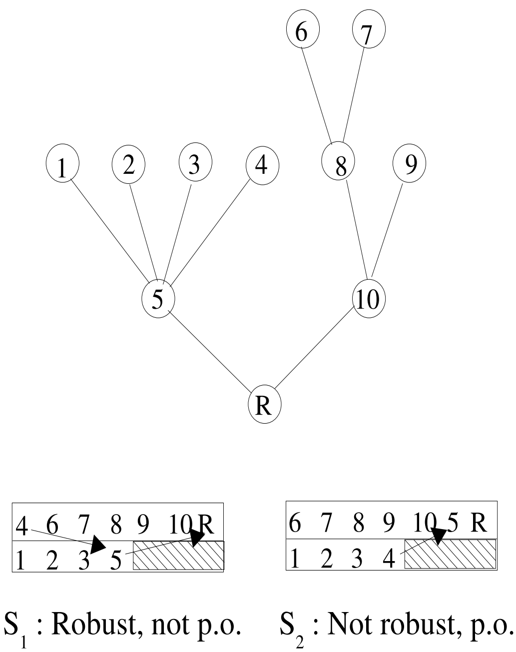



Figure 3: No dominance of PO-schedules for robustness in the LCT case

The example above, and the remark, show that the value of $\overline{c}$ and the structure of $c$ itself are important when comparing the robustness of two schedules (value of $\overline{c}$ had no influence in the SCT case). In the example, the first schedule is clearly more robust for $\overline{c} = 2$, but the situation is the opposite for, say, $\overline{c} = 5$. Finally, if $\overline{c}$ is unbounded, the only robust schedule is the trivial one. Among non trivial schedules, all UECT optimal PO-schedules have a deviation of at most $\overline{c} - \underline{c}$, and the deviation of any non PO-schedule is unbounded. Hence only PO-schedules can optimize robustness.

Searching for an algorithm optimizing the robustness when $c$ is an arbitrary vector in a fixed interval $[\underline{c}, \overline{c}]$ is still an open problem, and the complexity of such algorithm an open question. Still, CBOS has a small absolute sensitivity bound (our conjecture is that it is impossible to find an algorithm with a better bound) which is a good argument to call it robust in a weaker sense. This is not the case for ETF/LMS for large communication delays.

# 6 Conclusion

The 2-machine example shows the large difference of robustness between different algorithms, all optimal in the static case. Our belief is that such situation is quite common when scheduling under uncertainty, hence similar studies should prove useful before choosing an algorithm. Moreover, even for such a simple problem, optimizing the robustness can be a challenging problem. This difficulty is enforced by the influence of the definition of robustness itself. Last, our examples show that an algorithm can be robust for some uncertainty domain, but not for another one. Using such study necessitates a careful model of uncertainties.

When scheduling with uncertain communication delays, it is logical that processor-ordered schedules have good performances, as they limit the number and the impact of communications. This intuition is largely confirmed by the presented results. These results are due for a part to the optimality of CBOS algorithm. Of course PO-schedules generalize to $m$ machines (see [13, 18]), but the results, regarding dominance for instance, are more difficult to get, and less conclusive. The impossibility to present polynomial, optimal in deterministic case algorithms, might explain this.

The case of processing times also subject to uncertainty was not within the scope of this paper. However, any algorithm will be less sensitive in that case, as all actual processing times must be taken into account when computing the makespan of any schedule (whereas the amount of actual communication delays depend on the computed schedule).

# References

[1] J-C. Billaut, A. Moukrim, and E. Sanlaville (eds). *Flexibilité et Robustesse en Ordonnancement*. traité IC2, Hermes, 2005. in French, currently under translation.

[2] Ph. Chrétienne. A polynomial algorithm to optimally schedule tasks on a virtual distributed system under tree-like precedence constraints. *European Journal of Operational Research*, 43:225–230, 1989.

[3] R.L. Daniels and P. Kouvelis. Robust scheduling to hedge against processing time uncertainty in single stage production. *Management Science*, 41(2):363–376, 1995.

[4] F. Guinand, A. Moukrim, and E. Sanlaville. Sensitivity analysis of tree scheduling on two machines with communication delays. *Parallel Computing*, 30:103–120, 2004.

[5] F. Guinand, C. Rapine, and D. Trystram. Worst case analysis of lawler's algorithm for scheduling trees with communication delays. Technical report, LMC IMAG, Grenoble, France, 1995.

[6] F. Guinand and D. Trystram. Optimal scheduling of uect trees on two processors. *RAIRO Operation Research*, 34:131–144, 2000.

[7] N.G. Hall and M.E. Posner. Sensitivity analysis for scheduling problems. *Journal of Scheduling*, 7:49–83, 2004.

[8] C. Hanen and A. Munier. Performance of coffman graham schedule in the presence of unit communication delays. *Discr. Appl. Math.*, 81:93–108, 1998.

[9] W. Herroelen and R. Leus. Project scheduling under uncertainty, survey and research potential. *European Journal of Operational Research*, 2003. invited paper.

[10] J.J. Hwang, Y.C. Chow, F.D. Anger, and C.Y. Lee. Scheduling precedence graphs in systems with interprocessor communication times. *SIAM J. Comput.*, 18(2):244–257, 1989.

[11] E.L. Lawler. Scheduling trees on multiprocessors with unit communication delays. In *Workshop on Models and Algorithms for Planning and Scheduling Problems*, Villa Vigoni, Lake Como, Italy, 1993.

[12] V. Lebedev and I. Averbakh. Complexity of minimizing the total flow time with interval data and minmax regret criterion. *Discr. Appl. Math.* à paraître, special issue CO02.

[13] R. Lepere and D. Trystram. A new clustering algorithm for scheduling with large communication delays. In *IPDPS'02, 16th IEEE-ACM annual International Parallel and Distributed Processing Symposium*, Fort Lauderdale, Florida, 2002.

[14] A. Mahjoub. *Etude de la Robustesse et de la Flexibilité pour des problèmes d'ordonnancment et de localisations*. PhD thesis, INPG, Grenoble - France, juillet 2004.

[15] R.H. Möhring and M.K. Schäffter. A simple approximation algorithm for scheduling forests with unit processing times and zero-one communication delays. Technical Report 506, Technische Universität Berlin, 1995.

[16] B. Penz, C. Rapine, and D. Trystram. Sensitivity analysis of scheduling algorithms. *European Journal of Operational Research*, 134:606–615, 2001.

[17] C. Picouleau. *Etude des problèmes d'optimisation dans les systèmes distribués*. PhD thesis, Université Paris 6, France, 1992.

[18] E. Sanlaville. Sensitivity bounds for parallel machines with uncertain communication delays. *Journal of Scheduling*, 8:461–473, 2005.

[19] M. Veldhorst. A linear time algorithm for scheduling trees with communication delays optimally on two machines. Technical Report RUU-CS-93-04, Dept of Computer Sciences, Utrecht, Hollande, 1993.

# A practical approach for robust and flexible vehicle routing using metaheuristics and Monte Carlo sampling

Kenneth Sörensen*, Marc Sevaux†

**Abstract**

In this paper, we investigate how robust and flexible solutions of a number of stochastic variants of the capacitated vehicle routing problem can be obtained. To this end, we develop and discuss a method that combines a sampling based approach to estimate the robustness or flexibility of a solution with a metaheuristic optimization technique. This combination allows us to solve larger problems with more complex stochastic structures than traditional methods based on stochastic programming. It is also more flexible in the sense that adaptation of the approach to more complex problems can be easily done. We explicitly recognize the fact that the decision maker's risk preference should be taken into account when choosing a robust or flexible solution and show how this can be done using our approach.

**Key words :** stochastic vehicle routing, robustness, flexibility, Monte Carlo sampling, metaheuristics, memetic algorithm

## 1 Robust and flexible vehicle routing

The objective of vehicle routing problems is to determine the order in which to visit a spatially distributed set of customers using a set of vehicles so that the total travel cost (distance, time) is minimized. Standard vehicle routing formulations however, assume that all data concerning customer demand, travel costs, etc. are known with perfect certainty at design time. For many reasons, e.g. the uncertainty related to traffic conditions, these assumptions are unwarranted in a large number of practical situations. As a result,

*University of Antwerp, Faculty of Applied Economics. kenneth.sorensen@ua.ac.be
†University of South-Brittany, LESTER, CNRS FRE 2734. marc.sevaux@univ-ubs.fr

a number of vehicle routing formulations designed to find *robust* solutions have been developed. Three types of such formulations, commonly referred to as *stochastic vehicle routing problems* can be distinguished: problems with stochastic *travel times* (e.g. Lambert et al. [1993]), problems with stochastic *demands* (e.g. Stewart and Golden [1983]), and problems with stochastic *customers*, i.e. in which each customer has a certain probability of requiring service (e.g. Bertsimas [1992]). For a survey, we refer to Gendreau et al. [1996].

For stochastic problems, *robust* or *flexible* solutions are required. Consistent with the definitions found in the literature (see e.g. Kouvelis and Yu [1997], Mulvey et al. [1995], Roy [1998], Vincke [1999], Dias and Clímaco [1999], Branke [2001]), we call a solution to a stochastic optimization problem *robust* if it has a high quality regardless of the actual realization of the stochastic parameters. In other words, a solution that consistently has a high performance across all possible outcomes of the stochastic parameters, is called robust. Usually, stochastic optimization algorithms attempt to find the solution that has either the best *expected* quality or the best *worst-case* quality over all potential outcomes of the stochastic parameters. In many cases however, a solution can be (partially) changed once the actual values of the stochastic parameters become known. In this case, a solution is preferred that can be successfully adapted to any realization of the stochastic parameters. Such a solution is called *flexible*. In other words, a *flexible* solution is one that has a high quality after adaptation to the outcomes of the stochastic parameters, whereas a *robust* solution is one that has a high quality without adaptation to the stochastic parameters. In the remainder, we will call the procedure that is used to adapt the solution to the actual values of the stochastic parameters a *repair* procedure.

Several problems arise with these methods, that inhibit their use in real-life vehicle routing applications. First, most of the methods are *exact* methods, implying that they are unable to solve problems of realistic size. Indeed, as indicated by Birge [1997], the complexity of stochastic programs grows proportionally to the number of possible realisations of the stochastic parameters, which in turn grows exponentially with the number of stochastic parameters. Because in realistic cases this number is usually very large or even infinite (in the case of continuous distributions), only very small problems can be solved to "optimality". To partially overcome this problem, a part of the stochastic programming literature has fused on methods that use some form of *Monte Carlo sampling* of the stochastic parameters. For stochastic linear programming problems, *stochastic decomposition* [Higle and Sen, 1991] and *importance sampling* [Infanger, 1994] have been developed. For discrete stochastic problems, Norkin et al. [1998a,b] develop a sampling branch-and-bound. Whereas these methods use sampling at different steps of the optimization method to estimate e.g. function values, the *sample average approximation method* [Kleywegt et al., 2001] uses sampling to generate a set of sample scenarios and then attempts to optimize the corresponding deterministic expected-value problem. This method has been successfully applied to supply chain design problems [Goetschalckx et al., 2001] and routing problems [Verweij et al., 2003], among others.

A second drawback to traditional stochastic programming methods is the fact that they make heavy use of the specific stochastic structure of the problem they consider and are very difficult to adapt to other problems. Thirdly, almost all algorithms developed for stochastic routing attempt to find the solution with the optimal *expected value*. Although the average performance of a solution is certainly an important indication of its robustness, it does not take into account the risk-preference of the decision maker.

In this paper, we develop a method that is able to overcome these problems. We adopt a pragmatic approach to stochastic optimization and approximate the stochastically optimal solution on two levels. First, we use a sampling-based approach to quickly estimate the robustness of a solution. This allows us to quickly and simultaneously evaluate both average and worst-case performance of a solution, but more complicated measures of robustness can be incorporated. Our sampling-based approach also allows us to evaluate solutions of problems with many stochastic parameters. Secondly, we use a metaheuristic to find the solution with the best approximate robustness characteristic, which allows us to solve large problems. Our approach adapts a metaheuristic for a deterministic problem by replacing its evaluation function by a so-called *robust evaluation function*.

## 2 Problem description and deterministic solution method

The CVRP is defined on an undirected graph $G = (V, E)$ with a set of nodes $V = \{0, 1, \dots, n\}$. Node 0 corresponds to the depot, that has a set of identical vehicles of capacity $Q$ and maximal travel cost $C$. Nodes 1 to $n$ represent a set of spatially distributed customers, the demand of which is given by $q_i$. The travel cost between customer $i$ and customer $j$ is given by $c_{ij}$, the weight of the edge between node $i$ and node $j$. The objective of the deterministic VRP is to find a set of minimum total cost routes that have the following properties: (1) each route begins and ends at the depot, (2) each customer is visited exactly once, (3) the capacity and maximal travel cost of the vehicles is not exceeded.

To solve this problem, we have developed a MA|PM, or menetic algorithm with population management. A MA|PM is a memetic algorithm (a GA hybridised with local search) that uses distances to measure and control the diversity of a small population. This allows to maintain diversity in the population while keeping the size of the population small. For a more elaborate description of MA|PM, we refer to Sörensen and Sevaux [2005].

In our MA|PM, solutions are represented as a string of customers, *without trip delimiters* $S = S_1 S_2 \dots S_n$. When necessary (e.g. to calculate their objective function value) they are decoded optimally using the *split decoding procedure* by Prins [2004]. This procedure finds the optimal points at which to split the solution into tours so that the resulting decoded solution is feasible and the total travel cost is minimized. It makes use of an auxiliary directed weighted graph $H = (X, F, W)$. The vertex set $X$ contains $n + 1$ nodes (where $n$ is the number of customers served), labelled from 0 (the depot) to $n$.

The edge set $F$ is constructed as follows. An edge $(i, j)$ from vertex $i$ to vertex $j$ $(i < j)$ is added when a trip containing customers $S_{i+1}$ to $S_j$ is feasible, i.e. if

$$\sum_{k=i+1}^{j} q_{S_k} \leq Q \tag{1}$$

and

$$w_{ij} = c_{0S_{i+1}} + \sum_{k=i+1}^{j-1} c_{S_k S_{k+1}} + c_{S_j 0} + \sum_{k=i+1}^{j} e_{S_k} \leq C. \tag{2}$$

The weight $w_{ij}$ of the arc $(i, j)$ is equal to the sum of the travel distances covered in the trip $0 \rightarrow S_{i+1} \rightarrow S_{i+2} \rightarrow \ldots \rightarrow S_j \rightarrow 0$, plus the sum of drop costs in this trip. The minimum-cost path from vertex 0 to vertex $n$ in $H$ gives the shortest set of trips corresponding to this encoding. If more than one min-cost path exists, each of these paths will give a set of trips with equal total length. Since the graph $H$ contains only positive weights and no circuits, the shortest path can be efficiently calculated using Bellman's algorithm [Bellman, 1958].

A second procedure used heavily by our MA|PM for the CVRP is a distance measure based on the edit distance. Given three possible edit operations (add character, remove character and substitute character), the edit distance $d(s, t)$ is defined as the minimal number of edit operations required to transform a string $s$ into another string $t$. The edit distance is heavily studied in the literature. A simple dynamic programming algorithm [Wagner and Fischer, 1974] calculates this measure in $O(n^2)$ where $n$ is the length of both strings. Other, more efficient algorithms have been developed [Sörensen, 2003].

Vehicle routing solutions can be regarded as sets of strings, each string representing a tour. As tours have no direction, each string may be reversed. The distance measure for the CVRP developed in Sörensen [2003] calculates the minimal number of edit operations required to transform a VRP solution into another one, keeping into account that both solutions consist of a set of reversable strings. This is done by assigning tours from the first solution to the second one so that the total number of required edit operations is minimized.

Given two solutions $s$ and $t$ having $m$ and $n$ trips respectively, the distance measure constructs a square matrix $M$ of size $\max(m, n)$. Each column and each row represents a trip of $s$ and $t$ respectively. Empty trips are added to the solution with the smallest number of trips to give both solutions an equal number of trips. Cell $(i, j)$ of $M$ contains the minimum of $d(s_i, t_j)$ and $d(s_i, \tilde{t}_j)$, where $s_i$ is the $i$-th trip of solution $s$ and $\tilde{t}_j$ is the reversed $j$-th trip of solution $t$. A linear assignment problem is then calculated to find the matching of trips in solution $s$ to trips in solution $t$. The cost of the assignment problem is the distance between the two solutions.

In the MA|PM, the distance measure is used for *population management* in order to maintain the diversity of a small population of high-quality individuals and overcome problems of premature convergence. Before a (candidate) solution is added to the population, the algorithm checks to see whether it satisfies the *population diversity criterion*, i.e. whether it is sufficiently different from all other members of the population. This is done by measuring the so-called *distance to the population*, i.e. the minimum distance of the candidate solution to any solution already in the population. The solution cannot be added until its distance to the population is greater than or equal to the value of the *population diversity parameter* $\Delta$. This value can be used to control the diversity of the population as an increase of $\Delta$ will tend to increase population diversity, while a small value will tend to decrease it. Intensification and diversification phases can be alternated by setting the value of $\Delta$ to appropriate levels [Sörensen and Sevaux, 2005].

The MA|PM also uses a fast local optimization algorithm, that consists of two simple tabu search procedures that are used alternatively until neither of them finds any more improvements. The *insert* tabu search procedure attempts to locate any customer at any other potential location in the solution (i.e. any other location in any other tour) and takes the move that decreases the total distance most (or increases it least when no improving moves can be found). An insertion of a customer into another location is only allowed when no constraints are violated by this move and when the customer does not appear on the tabu list. This insert procedure is repeated until the best-found solution during this tabu search phase has not been improved for a fixed number of moves. A fixed-length tabu list is maintained to avoid cycling. The tabu list contains the customers that were most recently moved. The *swap* tabu search procedure attempts to swap every pair of customers and swaps the pair that yields the largest improvement in objective function value. The swap of a pair of customers is only allowed when the resulting solution does not violate any constraints and when the pair of customers does not appear on the tabu list. This procedure is repeated until a fixed number of non-improving moves. In this case, the tabu list contains pairs of of items that were most recently swapped. Pairs that appear on the tabu list are excluded from swapping.

The crossover operator used in our MA|PM is the *linear ordered crossover* (LOX). This operator starts by randomly dividing both parents into three parts. The first offspring solution is found by copying the middle part from parent one and completing the solution by circularly scanning parent two, starting from the third part and wrapping to the first part. The second offspring is formed by reversing the roles of the parents.

Our MA|PM for the CVRP starts by generating a set of random permutations of all customers. These are then decoded (using the split procedure) and subjected to local search (alternating the two tabu search operators until no more improvement is found). The initial population consists of a small number (usually 10) solutions. At each generation, two solutions are drawn from the population using a binary tournament selection operator (i.e. we retain the best of two randomly chosen solutions). These two solutions are subjected to the LOX operator. Both offspring solutions are then decoded and subjected to the tabu search local searches. When the solution satisfies the diversity criterion, it is added to the population. If not, it is *mutated* until it does satisfy the diversity criterion and added to the population. The mutation operator randomly swaps two customers and repeats this step until the resulting solution satisfies the diversity criterion.

We use an *adaptive* population management scheme, in which the value of $\Delta$ is slowly decreased to allow it to intensify and increased again when the search appears to be stuck in a local optimum, i.e. no more improving solutions are found for a fixed number of generations.

Some experiments show that our MA|PM is competitive with the best-known approaches in the literature. For detailed results, we refer to Sörensen [2003].

## 3 A sampling-based approach for robust and flexible vehicle routing

In this section, we modify the MA|PM developed in the previous section, so that it is able to deal with stochastic vehicle routing problems. This is done by replacing the objective function by a so-called *robust evaluation function*, that measures the robustness or flexibility of the solution. A *repair function* is used if the solution can be adapted when the actual values of the stochastic parameters become known.

In a nutshell, our approach is the following. We use a metaheuristic, such as our MA|PM, that generates a set of solutions that are both diverse and have a high quality. For each solution we encounter, we calculate one or more measures of robustness or flexibility, referred to as a robust evaluation function value. Finally, we choose the solution that has the best robust evaluation function value. When two or more measures of robustness or flexibility are calculated, we choose the solution that has the best combined value using a multi-objective decision making process.

A robust evaluation function value for a given solution is calculated by repeatedly applying the solution to a sampling of the stochastic parameters and calculating the corresponding (deterministic) objective function value. These objective function values are then combined into one or more measures of robustness. If a repair procedure is available, the solution is repaired before it is evaluated. The concept of repair function is closely related to the stochastic programming concept of *recourse*, but differs from it that any procedure, even a heuristic one, may be used.

A typical robust evaluation function $f^*$ is of the form

$$f^*(x) = \frac{1}{n_e} \sum_{i=1}^{n_e} f(x, \mathcal{S}_i(\pi)),$$

where $f$ is the (deterministic) objective function, $\pi$ is the set of stochastic parameters, and $\mathcal{S}$ is the sampling function. $\mathcal{S}_i(\pi)$ represents the $i$-th sampling of the stochastic parameters and $f(x, \mathcal{S}_i(\pi))$ is the objective function value of solution $x$ when applied to the $i$-th sampling. $n_e$ is the number of deterministic evalutions per robust evaluations, i.e. the number of evaluations of the solution on a random sampling of the stochastic parameters. If the solution can be repaired, the repair function $\mathcal{R}$ is invoked before evaluation the solution and the robust evaluation function becomes

$$f^*(x) = \frac{1}{n_e} \sum_{i=1}^{n_e} f(\mathcal{R}(x, \mathcal{S}_i(\pi))).$$

Other useful robust evaluation functions include the *worst-case* performance, given (for a minimization problem) by

$$f^*_{\text{wc}}(x) = \max_i f(x, \mathcal{S}_i(\pi)),$$

or the standard deviation

$$\sigma^*(x) = \sqrt{\frac{1}{n_e - 1} \sum_{i=1}^{n_e} [f(x; \mathcal{S}_i(\pi)) - f^*(x)]^2}.$$

These measures explicitly incorporate the risk preference of the decision maker into the process. E.g. the solution that has the best worst-case performance will generally be more "conservative" or "risk-averse" than the one that has the best average-case performance. The decision maker may decide to find the solution that minimizes $f^*(x) + \lambda\sigma^*(x)$, where $\lambda$ is a parameter expressing the risk-averseness of the decision maker. Other, even more complex measures of robustness may be used, such as the probability that the quality of the solution falls below a certain threshold. Of course, since they are obtained using Monte Carlo simulation, the robust evaluation function values are only estimates of the true robustness measures. However, statistical theory can be used to estimate the reliability of the estimates. For a sufficiently large number of evaluations (e.g. $n_e > 30$), the central limit theorem states that $f^*(x)$ is approximately normally distributed. From this, it follows that an $100(1 - \alpha)$ confidence interval for the real average performance of a solution is given by

$$f^*(x) \pm z_{1-\frac{\alpha}{2}} \sqrt{\frac{(\sigma^*(x))^2}{n_e}}.$$

For a more elaborate discussion, we refer to Law and Kelton [1999].

One possible extension of our method is the use of *penalty functions* to penalize violation of constraints. For some samples of the stochastic parameters, a solution might become infeasible. A penalty function $\mathcal{P}$ may be used to express the severity of the constraint violations and added to the robust evaluation function value as follows:

$$f^*(x) = \frac{1}{n_e} \sum_{i=1}^{n_e} [f(x; \mathcal{S}_i(\pi)) + \mathcal{P}(x; \mathcal{S}_i(\pi))].$$

In our method, stochastic information can be expressed both as independent probability distributions on the parameters of the problem, or as a set of scenarios in which each of the parameters has a fixed value. In the latter case, the value of $n_e$ is the number of scenarios, and the sampling function $\mathcal{S}$ is used to evaluate the performance of a solution under the different scenarios.

We believe that our approach has several advantages over traditional methods based on stochastic programming, most importantly the fact that it is considerably more flexible, considerably easier to implement and applicable to much larger and much more complex problems. Moreover, it allows the decision maker to evaluate complex robustness characteristics of a solution, using any type of stochastic information that is available and incorporating his risk preference.

# 4 Experiments

In this section, we perform some experiments on different stochastic versions of the CVRP. Unless otherwise indicated, our MA|PM uses a population of size 10. The population diversity parameter $\Delta$ is set to 20 initially and reduced by one unit for each 3 generations. $\Delta$ is set to its initial level after 60 unproductive generations. Both the *swap tabu search* and the *move tabu search* procedures use a tabu tenure of 30 and repeat until 10 -improving moves.

Data sets for the stochastic versions are derived from the Christofides et al. [1979] instances for the deterministic CVRP, available from the OR Library[1]. We generally assume that the distributions of the stochastic parameters are independent and have their mean equal to the corresponding value in the deterministic data set. Unless otherwise indicated, the robust evaluation function value is based on $n_e = 1000$ evaluations.

[1] `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/vrpinfo.html`

## 4.1 Vehicle routing with stochastic demand and cost

In this set of experiments, customer demand $q_i$ is assumed to be uniformly distributed between $0.75\bar{q}_i$ and $1.25\bar{q}_i$. Travel cost between customers $i$ and $j$ are uniformly distributed between $0.8\bar{c}_{ij}$ and $1.2\bar{c}_{ij}$. $\bar{q}_i$ and $\bar{c}_{ij}$ are the averages of the demand and travel cost distributions respectively and equal to the values found in the deterministic data set. Note that the deterministic problem is the *mean value formulation* of the stochastic problem, i.e. the deterministic problem can be obtained by setting all stochastic parameters to their expected value.

We assume that the routes have to be determined before the actual values of customers demand and travel cost are known and cannot be changed afterwards. We therefore attempt to find *robust* solutions. For some samples of the stochastic demand and cost, a solution might violate the maximum travel cost constraints per route or the capacity constraints of the vehicles. We therefore introduce two penalty functions. Let $c_{ijk}$ and $q_{ik}$ represent the $k$-th random numbers taken from the distributions of $c_{ij}$ and $q_i$ respectively. The deterministic objective function value of solution $x$, evaluated on the $k$-th sampling of the stochastic parameters can be calculated as follows.

$$f(x; \mathcal{S}_k(\pi)) = \sum_l \sum_{e_{ij} \in E_l(x)} c_{ijk},$$

where $E_l(x)$ represents the set of edges included in the $l$-th tour of solution $x$, $e_{ij}$ the edge between customers $i$ and $j$. If the total cost in a given route is larger than $C$ (the maximum cost), then a fixed penalty per unit of exceeded cost is added.

$$\mathcal{P}_{\text{cost}}(x; \mathcal{S}_k(\pi)) = \alpha_1 \sum_l \max(0, \sum_{e_{ij} \in E_l(x)} c_{ijk} - C).$$

Similarly, if the total demand served in a given route exceeds the maximum demand $Q$, a fixed penalty cost per unit of exceeded demand is added.

$$\mathcal{P}_{\text{capacity}}(x; \mathcal{S}_k(\pi)) = \alpha_2 \sum_l \max(0, \sum_{v_i \in V_l(x)} q_{ik} - Q), \tag{3}$$

where $V_l(x)$ represents the set of vertices (customers) in tour $l$ of solution $x$. $\alpha_1$ is the penalty cost per unit of exceeded travel cost. $\alpha_2$ is the penalty per unit of exceeded capacity. We set $\alpha_1 = 100$ and $\alpha_2 = 500$.

We look for solutions that have both a good average-case and a good worst-case performance. To this end, we define two robust evaluation functions: $f^*(x)$ and $f^*_{\text{wc}}$. To calculate $f^*$ for solution $x$, we calculate the total travel cost increased with the values of both penalty functions for $n_e$ random samples of the stochastic parameters and take the average. For $f^*_{\text{wc}}$, we take the maximum value. For each solution encountered, we also measure the standard deviation $\sigma^*(x)$ and the value $f(x)$, the deterministic objective function value.

The MA|PM is applied to the 14 vehicle routing problems. Robust evaluation function $f^*$ is used in the binary tournament selection procedure to select solutions from the population for crossover. This guides the search towards more robust solutions. We stop the MA|PM after 200 generations, yielding 400 solutions. Detailed results are in appendix (table3). Table1 is an extract from this table holding the results of a single experiment.

| **Data file** | $n$ | **Criterion** | $f$ | $f^*$ | $\sigma^*$ | $f^*_{\text{wc}}$ |
|---|---|---|---|---|---|---|
| vrpnc01 | 50 | $f$ | 549.76 | 3211.60 | 2891.00 | 17779.56 |
| | | $f^*_1$ | 604.76 | 605.01 | 13.21 | 876.66 |
| | | $f^*_2$ | 629.12 | 629.28 | 10.61 | 661.48 |

Table 1: Vehicle routing with stochastic demand and costs, example result

For each data set, three rows are presented. The first row contains the characteristics of the solution that minimizes $f$, i.e. the solution that would be found by our method in the deterministic case, if we would ignore all stochastic information. The second and third rows contain the characteristics of the solution that minimizes $f^*$ and $f^*_{\text{wc}}$ respectively. It can be seen that the solutions in the second and third rows score considerably better with respect to all robustness indicators ($f^*$, $f^*_{\text{wc}}$ and $\sigma^*$). These two solutions are therefore considerably more robust than the solution in the first row, proving that their is a strong need to take the stochastic information into account.

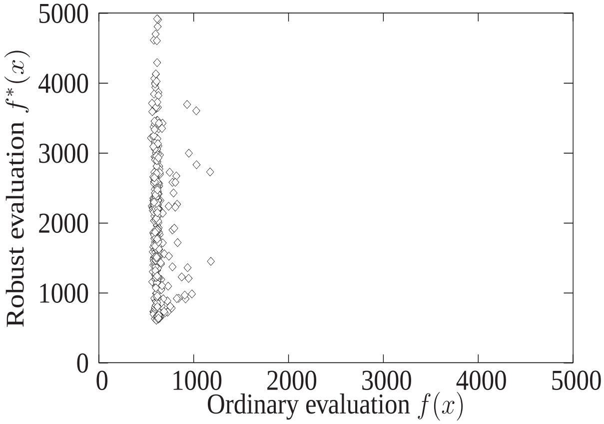



Figure 1: Ordinary evaluation and robust evaluation for vrpnc01

This is further illustrated in figure1, where we plot the values of $f$ versus $f^*$ for all 400 solutions. From this figure, it is clear that all robust solutions (low $f^*$) have a good deterministic objective function value, but that the reverse is not necessarily true. The decision maker can choose on of the two robust solutions presented in table3 or alternatively one of the other non-dominated solutions (solutions for which there does not exist a solution which scores better with respect to all robust evaluation functions) generated by the MA|PM. These are not shown because of space restrictions.

## 4.2 Vehicle routing with stochastic customers and a repair function

In some situations, the vehicle dispatcher does not know in advance which customers will require service and which not. A possible strategy is to route all customers and repair the solution once the list of customers that do require service becomes known. This situation may occur when customers need to book a service in advance, but may cancel at short notice. A solution that can be effectively repaired is called *flexible*.

In an experiment, each customer has a 50% probability of requiring service. A 100% service level is required and the company therefore decides to design conservative solutions that are still feasible if all customers in a route require service. The repair procedure used by the company is to remove all customers that do not require service from a given route. If the customer in $i$-th position in a given route is removed, the vehicle travels directly from the customer in position $i-1$ to the customer in position $i+1$. Total travel cost is measured after the customers have been removed from the routes.

The MA|PM is run for 200 generations. Each of the 400 generated solutions is evaluated 1000 times on a random list of customers. As in the previous experiment, we record the average $f^*$, the maximum $f^*_{\text{wc}}$ and the standard deviation $\sigma^*$ over all $n_e$ evaluations. Results can be found in table4.

From this table, it is clear that not much profit can be gained by using the robust optimization approach, as the solution that has the best value of $f$ has approximately the same robustness characteristics as the solution that optimizes one of the robust evaluation function values. The main reason for this is the requirement that the solution should be feasible even when all customers are present. This results in very conservative solutions that—when the actual customers to serve are observed—use only a fraction of the available resources (capacity and maximum cost). The best solution of the deterministic problem, utilising as much of these resources as possible, is therefore likely to be a better solution for the reduced set of customers, than other solutions that do not use the resources as fully. As a result, this solution is very likely to be flexible.

# 5 Use of the robust evaluation function and computation times

The use of our approach for robust and flexible vehicle routing requires some design issues to be tackled. Especially the question when to use the robust evaluation function, should be taken with some care. Obviously, the robust evaluation function is used to evaluate the robustness properties of a solution. However, it can also be used to guide the search towards more robust solutions. In our MA|PM, the robust evaluation function is used to guide the search in the binary tournament selection process. However, it is not used for the move selection of the tabu search local optimization procedures. In both tabu search procedures, the (deterministic) objective function value is used to select the next move. The reason for this is that the use of the robust evaluation function for move selection in the tabu search procedures would have resulted in prohibitive computing time increases. The bulk of computing time is spent on improving solutions using the tabu search procedures. While this is necessary to obtain high quality solutions, the tabu search procedures require a large number of evaluations. Moreover, the calculation of the cost of a move can easily be short-circuited when a deterministic objective function is used, but this no longer holds when a robust evaluation function is used. Indeed, it can be easily seen that that when a customer $k$ is inserted between customers $i$ and $j$, the total cost of this route changes by $c_{ik}+c_{kj}-c_{ij}$ (provided that the route remains feasible). Similar reasoning applies for removing a customer from a route or swapping two customers. These shortcut calculations allow a move to be evaluated without re-evaluating the entire solution, but they are no longer valid for a robust evaluation function.

By using the robust evaluation function for selection of solutions from the population, it only has to be calculated twice for each generation (when both solutions are added to the population). This results in relatively small computing time increases as can be seen in table 2. This table compares the computing times for 200 generations for the MA|PM in the deterministic and the stochastic case (experiment with stochastic demand and cost). Computation time never increases by more than a factor of 3.08.

One of the design choices that influence the computation time when using our robust optimization approach is the value of $n_e$, the number of objective function evaluations to perform to calculate the value of the robust evaluation function. In the experiments performed in this paper, we put the value of $n_e$ at 1000 and still obtained relatively small computing time increases, but this may not always be the case. Especially in situations where the objective function is time-intensive to calculate, the value of $n_e$ must be carefully chosen. As we mentioned before, a confidence interval can be calculated around the value of $f^*$. The value of $n_e$ should be chosen in such a way that the confidence interval is small enough. How small exactly the confidence interval should be, depends on the specific application, the level of confidence that is required by the decision maker and the computing time available.

| **Data file** | $n$ | **avg. determ. (s)** | **avg. stoch. (s)** | $\frac{\text{avg. stoch.}}{\text{avg. determ.}}$ |
|---|---|---|---|---|
| vrpnc01 | 50 | 68.84 | 99.30 | 1.44 |
| vrpnc02 | 75 | 205.59 | 408.59 | 1.99 |
| vrpnc03 | 100 | 363.23 | 573.17 | 1.58 |
| vrpnc04 | 150 | 932.70 | 1856.12 | 1.99 |
| vrpnc05 | 199 | 1969.29 | 4266.21 | 2.17 |
| vrpnc06 | 50 | 92.57 | 97.52 | 1.05 |
| vrpnc07 | 75 | 205.43 | 423.80 | 2.06 |
| vrpnc08 | 100 | 369.90 | 655.88 | 1.77 |
| vrpnc09 | 150 | 967.54 | 2329.98 | 2.41 |
| vrpnc10 | 199 | 1657.46 | 5102.75 | 3.08 |
| vrpnc11 | 120 | 667.53 | 1215.21 | 1.82 |
| vrpnc12 | 100 | 292.09 | 698.23 | 2.39 |
| vrpnc13 | 120 | 772.81 | 1008.80 | 1.31 |
| vrpnc14 | 100 | 526.89 | 722.35 | 1.37 |
| All | N/A | 649.42 | 1389.85 | 2.14 |

Table 2: Computing times (200 generations)

## 6 Conclusions

In this paper, we have developed an approach to find robust and flexible solutions of several stochastic versions of the capacitated vehicle routing problem. Our approach combines the optimization power of metaheuristics with Monte Carlo simulation based estimation of the robustness or flexibility of a solution. A repair procedure is introduced when flexible solutions are needed and penalty functions may be introduced to penalize violation of constraints. Our approach recognizes the need for more complex expressions of robustness or flexibility than the often-used average performance to be entered into the decision-making process.

We have tested our approach on vehicle routing problems with stochastic demand and cost and with stochastic customers. We have also discussed some design issues, specifically how the search can be guided towards robust or flexible solutions and how the computing time increase can be kept within limits.

We believe that our approach offers considerable advantages over more traditional methods based on stochastic programming, especially when applied to large-scale, real-life stochastic vehicle routing applications.

## References

R. Bellman. On a routing problem. *Quarterly Journal of Applied Mathematics*, 16(1): 87–90, 1958.

D.J. Bertsimas. A vehicle routing problem with stochastic demand. *Operations Research*, 40:574–585, 1992.

J.R. Birge. Stochastic programming computation and applications. *INFORMS Journal on Computing*, 9:111–133, 1997.

J. Branke. *Evolutionary Optimization in Dynamic Environments*. Kluwer, Boston, 2001.

N. Christofides, A. Mingozzi, P. Toth, and C.Sandi. *Combinatorial optimization*. John Wiley, Chichester, 1979.

L.C. Dias and J. Clímaco. On computing ELECTRE's credibility indices under partial information. *Journal of Multi-Criteria Decision Analysis*, 8:74–92, 1999.

M. Gendreau, G. Laporte, and R. Séguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88:3–12, 1996.

M. Goetschalckx, S. Ahmed, A. Shapiro, and T. Santoso. Designing flexible and robust supply chains. In A. Artiba, editor, *Proceedings of the International Conference on Industrial Engineering and Production Management*, pages 539–551, Quebec, Canada, 2001.

J.L. Higle and S. Sen. Stochastic decomposition: An algorithm for two stage stochastic linear programs with recourse. *Mathematics of Operations Research*, 16:650–669, 1991.

G. Infanger. *Planning Under Uncertainty: Solving Large Scale Stochastic Linear Programs*. Boyd and Fraser Publishing Co., Danvers, MA, 1994.

A.J. Kleywegt, A. Shapiro, and T. Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12:479–502, 2001.

P. Kouvelis and G. Yu. *Robust Discrete Optimisation and its Applications*, volume 14 of *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers, Dordrecht, 1997.

V. Lambert, G. Laporte, and F.V. Louveaux. Designing collection routes through bank branches. *Computers and Operations Research*, 20:783–791, 1993.

A.M. Law and W.D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, London, 1999.

J.M. Mulvey, R.J. Vanderbei, and S.A. Zenios. Robust optimization of large-scale systems. *Operations Research*, 43:264–281, 1995.

V.I. Norkin, Y.M. Ermoliev, and A. Ruszczyński. On optimal allocation of indivisibles under uncertainty. *Operations Research*, 46:381–395, 1998a.

V.I. Norkin, G.Ch. Pflug, and A. Ruszczyński. A branch and bound method for stochastic global optimization. *Mathematical Programming*, 83:425–450, 1998b.

C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers and Operations Research*, 31(12):1985–20021, 2004. URL `doi:10.1016/S0305-0548(03)00158-8`.

B. Roy. A missing link in OR-DA: Robustness analysis. *Foundations of Computing and Decision Sciences*, 23:141–160, 1998.

K. Sörensen. *A framework for robust and flexible optimisation using metaheuristics with applications in supply chain design*. PhD thesis, University of Antwerp, Belgium, 2003.

K. Sörensen and M. Sevaux. MA|PM: memetic algorithms with population management. *Computers and Operations Research*, 33(5):1214–1225, 2005. URL `doi:10.1016/j.cor.2004.09.011`.

W.R. Stewart and B.L. Golden. Stochastic vehicle routing: A comprehensive approach. *European Journal of Operational Research*, 14:371–385, 1983.

B. Verweij, S. Ahmed, A.J. Kleywegt, G. Nemhauser, and A. Shapiro. The sample average approximation method applied to stochastic routing problems: A computational study. *Computational Optimization and Applications*, 24:289–333, 2003.

P. Vincke. Robust solutions and methods in decision aid. *Journal of Multi-Criteria Decision Analysis*, 8:181–187, 1999.

R.A. Wagner and M.J. Fischer. The string-to-string correction problem. *Journal of the Association for Computing Machinery*, 21:168–173, 1974.

Table 3: Vehicle routing with stochastic demand and costs

| **Data file** | $n$ | **Criterion** | $f$ | $f^*$ | $\sigma^*$ | $f^*_{\text{wc}}$ | **CPU (s)** |
|---|---|---|---|---|---|---|---|
| vrpnc01 | 50 | $f$ | 549.76 | 3211.60 | 2891.00 | 17779.56 | 101 |
| | | $f_1^*$ | 604.76 | 605.01 | 13.21 | 876.66 | |
| | | $f_2^*$ | 629.12 | 629.28 | 10.61 | 661.48 | |
| vrpnc02 | 75 | $f$ | 891.86 | 8335.36 | 4986.75 | 30836.52 | 349 |
| | | $f_1^*$ | 956.57 | 1134.44 | 617.56 | 6509.91 | |
| | | $f_2^*$ | 956.57 | 1134.44 | 617.56 | 6509.91 | |
| vrpnc03 | 100 | $f$ | 887.97 | 2904.39 | 2753.68 | 16150.60 | 541 |
| | | $f_1^*$ | 971.38 | 977.51 | 72.99 | 2236.30 | |
| | | $f_2^*$ | 971.38 | 977.51 | 72.99 | 2236.30 | |
| vrpnc04 | 150 | $f$ | 1149.53 | 5313.22 | 3869.58 | 23450.44 | 1893 |
| | | $f_1^*$ | 1208.36 | 1596.03 | 1063.23 | 9598.45 | |
| | | $f_2^*$ | 1208.36 | 1596.03 | 1063.23 | 9598.45 | |
| vrpnc05 | 199 | $f$ | 1512.77 | 10112.13 | 5737.19 | 33182.36 | 4153 |
| | | $f_1^*$ | 1687.18 | 3040.09 | 2118.64 | 15630.12 | |
| | | $f_2^*$ | 1687.18 | 3040.09 | 2118.64 | 15630.12 | |
| vrpnc06 | 50 | $f$ | 559.87 | 567.68 | 122.64 | 3350.03 | 98 |
| | | $f_1^*$ | 559.87 | 567.68 | 122.64 | 3350.03 | |
| | | $f_2^*$ | 579.51 | 579.42 | 9.58 | 605.79 | |
| vrpnc07 | 75 | $f$ | 995.55 | 2583.40 | 2285.52 | 11686.45 | 415 |
| | | $f_1^*$ | 1070.72 | 1077.22 | 79.39 | 2512.38 | |
| | | $f_2^*$ | 1070.72 | 1077.22 | 79.39 | 2512.38 | |
| vrpnc08 | 100 | $f$ | 945.33 | 946.03 | 38.49 | 2106.94 | 648 |
| | | $f_1^*$ | 945.33 | 946.03 | 38.49 | 2106.94 | |
| | | $f_2^*$ | 965.35 | 965.21 | 11.81 | 999.43 | |
| vrpnc09 | 150 | $f$ | 1352.35 | 1506.98 | 659.95 | 6979.06 | 2265 |
| | | $f_1^*$ | 1404.10 | 1404.68 | 16.24 | 1460.92 | |
| | | $f_2^*$ | 1404.10 | 1404.68 | 16.24 | 1460.92 | |
| vrpnc10 | 199 | $f$ | 1597.51 | 3683.01 | 2822.85 | 18401.49 | 5117 |
| | | $f_1^*$ | 1697.23 | 1698.18 | 31.61 | 2415.77 | |
| | | $f_2^*$ | 1784.14 | 1783.08 | 18.10 | 1837.73 | |
| vrpnc11 | 120 | $f$ | 1260.55 | 4775.36 | 3331.22 | 21620.35 | 1229 |
| | | $f_1^*$ | 1492.17 | 1504.12 | 212.15 | 6115.15 | |
| | | $f_2^*$ | 1507.27 | 1507.52 | 26.71 | 1601.84 | |
| vrpnc12 | 100 | $f$ | 883.55 | 8004.79 | 5372.87 | 30825.31 | 698 |
| | | $f_1^*$ | 1138.66 | 1638.14 | 1359.21 | 9534.97 | |
| | | $f_2^*$ | 1138.66 | 1638.14 | 1359.21 | 9534.97 | |
| vrpnc13 | 120 | $f$ | 1368.33 | 4629.15 | 3227.07 | 18598.48 | 1079 |
| | | $f_1^*$ | 1565.28 | 1611.20 | 315.01 | 6819.88 | |
| | | $f_2^*$ | 1727.20 | 1734.02 | 73.03 | 3117.71 | |
| vrpnc14 | 100 | $f$ | 952.46 | 7706.49 | 5126.85 | 29321.32 | 719 |
| | | $f_1^*$ | 1033.58 | 1775.53 | 1707.77 | 13073.67 | |
| | | $f_2^*$ | 1125.71 | 2192.62 | 1891.78 | 12757.21 | |

Table 4: Vehicle routing with stochastic customers and repair procedure

| **Data file** | $n$ | **Criterion** | $f$ | $f^*$ | $\sigma^*$ | $f^*_{\text{wc}}$ | **CPU (s)** |
|---|---|---|---|---|---|---|---|
| vrpnc01 | 50 | $f$ | 527.46 | 424.39 | 28.30 | 494.89 | 99.30 |
| | | $f_1^*$ | 528.22 | 423.92 | 28.85 | 488.88 | |
| | | $f_2^*$ | 528.22 | 423.92 | 28.85 | 488.88 | |
| vrpnc02 | 75 | $f$ | 905.22 | 736.65 | 45.95 | 854.83 | 408.59 |
| | | $f_1^*$ | 942.20 | 721.21 | 43.89 | 846.89 | |
| | | $f_2^*$ | 927.41 | 728.53 | 47.00 | 836.21 | |
| vrpnc03 | 100 | $f$ | 852.05 | 703.72 | 34.09 | 788.00 | 573.17 |
| | | $f_1^*$ | 852.14 | 696.88 | 32.00 | 776.41 | |
| | | $f_2^*$ | 852.14 | 696.88 | 32.00 | 776.41 | |
| vrpnc04 | 150 | $f$ | 1130.42 | 977.25 | 32.34 | 1067.67 | 1856.12 |
| | | $f_1^*$ | 1130.42 | 977.25 | 32.34 | 1067.67 | |
| | | $f_2^*$ | 1134.48 | 979.88 | 33.22 | 1065.78 | |
| vrpnc05 | 199 | $f$ | 1466.17 | 1286.28 | 32.03 | 1366.59 | 4266.21 |
| | | $f_1^*$ | 1483.99 | 1282.70 | 39.68 | 1384.81 | |
| | | $f_2^*$ | 1466.17 | 1286.28 | 32.03 | 1366.59 | |
| vrpnc06 | 50 | $f$ | 567.92 | 457.43 | 32.55 | 533.51 | 97.52 |
| | | $f_1^*$ | 577.01 | 456.09 | 34.14 | 553.61 | |
| | | $f_2^*$ | 569.51 | 459.30 | 32.16 | 528.00 | |
| vrpnc07 | 75 | $f$ | 990.71 | 809.90 | 51.78 | 919.41 | 423.80 |
| | | $f_1^*$ | 1000.02 | 806.21 | 56.00 | 945.15 | |
| | | $f_2^*$ | 990.71 | 809.90 | 51.78 | 919.41 | |
| vrpnc08 | 100 | $f$ | 933.72 | 786.23 | 36.69 | 872.24 | 655.88 |
| | | $f_1^*$ | 942.00 | 779.18 | 40.24 | 890.61 | |
| | | $f_2^*$ | 933.72 | 786.23 | 36.69 | 872.24 | |
| vrpnc09 | 150 | $f$ | 1351.54 | 1152.77 | 40.75 | 1257.38 | 2329.98 |
| | | $f_1^*$ | 1351.54 | 1152.77 | 40.75 | 1257.38 | |
| | | $f_2^*$ | 1351.54 | 1152.77 | 40.75 | 1257.38 | |
| vrpnc10 | 199 | $f$ | 1574.00 | 1379.91 | 41.69 | 1497.29 | 5102.75 |
| | | $f_1^*$ | 1577.60 | 1377.88 | 42.73 | 1483.68 | |
| | | $f_2^*$ | 1577.60 | 1377.88 | 42.73 | 1483.68 | |
| vrpnc11 | 120 | $f$ | 1122.20 | 1019.96 | 21.55 | 1083.10 | 1215.21 |
| | | $f_1^*$ | 1123.32 | 1019.26 | 22.61 | 1081.70 | |
| | | $f_2^*$ | 1125.26 | 1020.72 | 21.19 | 1074.73 | |
| vrpnc12 | 100 | $f$ | 861.27 | 774.77 | 24.27 | 839.91 | 698.23 |
| | | $f_1^*$ | 862.98 | 771.46 | 24.77 | 820.59 | |
| | | $f_2^*$ | 862.98 | 771.46 | 24.77 | 820.59 | |
| vrpnc13 | 120 | $f$ | 1239.06 | 1121.11 | 22.49 | 1177.47 | 1008.80 |
| | | $f_1^*$ | 1250.98 | 1117.50 | 23.57 | 1177.95 | |
| | | $f_2^*$ | 1239.06 | 1121.11 | 22.49 | 1177.47 | |
| vrpnc14 | 100 | $f$ | 878.46 | 768.03 | 23.92 | 833.47 | 722.35 |
| | | $f_1^*$ | 878.46 | 768.03 | 23.92 | 833.47 | |
| | | $f_2^*$ | 878.46 | 768.03 | 23.92 | 833.47 | |

# Recherche de conclusions robustes dans une problématique de placements financiers en environnement incertain

Philippe Vallin [1]

**Résumé**

Dans cette analyse, on cherche à exhiber des conclusions et solutions robustes dans un problème d'investissement sous contrainte budgétaire dans un environnement incertain.

La notion de robustesse est traduite par le critère de Savage. L'incertitude se traduit par la méconnaissance des rentabilités des investissements (ou placements), sur un horizon fixé. Celles-ci peuvent prendre des valeurs positives ou négatives. L'impact du critère sur le choix des politiques nous a paru intéressant à observer en comparaison des pratiques usuelles.

On obtient les conclusions suivantes :

1. Une solution optimale pour un jeu de rentabilité fixé n'est pas robuste

2. N'investir dans aucun placement n'est pas non plus une politique « robuste »

3. Lorsque le taux maximum de perte est supérieur ou égal au taux maximum de gain,

   a. La consommation totale du budget n'est pas une politique robuste, compte tenu du critère cela ne semble pas étonnant, mais,

   b. Pour obtenir une politique robuste, il faut investir dans les deux placements à plus faible coût (un seul ou éventuellement deux).

4. Bien que les coûts unitaires des placements soient supposés déterminés, il est inutile d'estimer le coût des placements les plus onéreux, ils n'interviennent pas dans la décision.

**Mots-clefs :** décision robuste, investissement, critère de Savage.

[1] Lamsade, Université Paris Dauphine, {vallin@lamsade.dauphine.fr}

**Abstract**

In this analysis, we seek robust conclusions and solutions in an investment problem under budgetary constraint and uncertainty.

The concept of robustness is given by the Savage criterion. Uncertainty comes from ignorance of profitability rates of the investments (or placements), on a fixed horizon. They can have positive or negative values. The impact of the criterion on the choice of the policies appears interesting comparing the usual practice. The following conclusions are obtained:

1. An optimal solution for a fixed set of profitability rates is not robust

2. No investment is not a "robust" policy

3. When the maximum rate of loss is higher than the maximum rate of profit,

   a. The overall consumption of the budget is not a robust policy, taking into account the criterion that does not seem astonishing, but,

   b. To obtain a robust policy, it is necessary to invest in the two cheaper placements (only one or both).

4. Although the unit costs of the placements are supposed to be given, it is useless to evaluate the cost of the most expensive placements; they are not taken into account in the decision.

**Key Words**: robust decision, investment, Savage criterion.

# 1 Contexte et modèle

On cherche à investir en choisissant dans un portefeuille de $n$ types ($i$ =1,…,$n$) de placements possibles (actions, obligations, d'emprunt d'Etat, immobilier …) en avenir incertain.

On dispose d'un budget disponible par convention égal à 1, ce qui ne nuit pas à la généralité.

On supposera que sur l'horizon de l'étude on peut multiplier par un facteur $r$ le capital investi ou perdre tout ou partie du capital.

La notion de robustesse est traduite par le critère de Savage (max du regret appliqué au gain net) que l'on minimise évidemment !

On peut par exemple se situer dans le cadre d'une institution financière qui gère des patrimoines pour le compte de ses clients. Le regret paraît assez naturel comme critère pour le client qui juge en fin d'horizon pour un état du marché donné : « ah ! si vous m'aviez conseillé cet investissement j'aurais gagné $x$ k€ de plus ! ». On peut donc concevoir qu'un gestionnaire prudent choisisse un portefeuille qui minimise le risque de regret en fonction des états possibles du marché, risque traduit par le regret maximum, c'est-à-dire le pire cas en fonction des évolutions des marchés.

Le critère est le gain ou la perte relatifs au capital investi, il s'exprime par :

$$Z = r_1 x_1 + r_2 x_2 + \ldots + r_n x_n,$$

où

_ $r_i$ , $-s \leq r_i \leq a$, est le taux de gain (ou perte) ; $a > 0$ sinon la solution consiste à ne pas investir puisque la perte est assurée ; $0 < s \leq 1$ est le seuil plancher de perte (s=1 signifie que la totalité du capital investi peut être perdu) , $a$ et $s$ sont supposés liés au marché il sont identiques pour tous les types d'investissements du portefeuille ; $a+s$ peut être considéré comme un indicateur de « volatilité ». Lorsque $r_i$ =+1, le capital est doublé, si $r_i$ = -1, le capital est perdu.

_ $x_i$, le montant du capital investi dans le placement $i$ ,exprimé en unité monétaire, comme le budget est 1 par convention, $x_i$ peut être interprétée comme une part du budget.

Au capital investi s'ajoute un coût, $c_i$, ou droit d'entrée (frais de courtage, droits de gardes, coût de gestion, taxe, droits d'enregistrement)[2] que nous considérons comme proportionnel au capital investi. Le montant total de l'investissement est donc

$$p_i x_i = (1+ c_i) x_i$$

La contrainte budgétaire s'exprime par :

$p_1 x_1 + p_2 x_2 + \ldots + p_n x_n, \leq 1$ (l'unité monétaire est le budget connu et fixé).

On suppose $p_1 < p_2 < \ldots < p_n$ , c'est-à-dire que les investissements sont classés dans l'ordre croissant des coûts : $c_1 < c_2 \ldots . < c_n$ , inégalités strictes car on ne distingue pas deux placements de même coût puisqu'on ne dispose d'aucune information discriminante sur les rendements. Sous cette hypothèse, on peut considérer que les $i$ représentent d'avantage des familles d'investissements que des investissements spécifiés.

La solution « robuste » (au sens de Savage) est dans ce contexte définie par :

$$\text{Min}_{x_1 x_2 .. x_n} \{\text{Max}_{r_1 r_2 \ldots r_n} (Z^*(r_1,..,r_n) - Z(x_1,..,x_n ; r_1,..,r_n))\}$$

$$\sum_i p_i x_i \leq 1$$

$$-s \leq r_i \leq a$$

[2] Le critère à optimiser pourrait être $Z = (r_1 - c_1)x_1 + \ldots + (r_n - c_n)x_n$ représentant le gain net. Nous négligeons ici l'impact de ces coûts dans le critère considérant qu'il est négligeable compte tenu de l'incertitude prise en compte.

où $Z^*(r_1,\ldots,r_n)$ est la solution optimale pour un ensemble de paramètres $(r_1,..,r_n)$ fixé.

# 2 Analyse

Le tableau classique des regrets avec les « actions possibles » en lignes et les « états de la nature » en colonnes s'exprime ici par une infinité de lignes et de colonnes puisque le choix des portefeuilles se caractérise par des attributions monétaires et les taux de rendement sont des paramètres continus. Mais on peut dans ce contexte se ramener à un cas discret en ce qui concerne les « états de la nature ».

**Lemme 1 :** Pour un portefeuille donné, la valeur du regret maximum ne met en jeu que $n$+1 valeurs de regrets.

**Preuve** :

En effet, quelles que soient les valeurs des $r_i$ , le gain optimal $Z^*(r_1,\ldots,r_n)$ ne s'exprime que par $n$+1 expressions distinctes. Ceci est lié à la simplicité de la solution pour un jeu de $r_i$ donné : il suffit de concentrer l'investissement sur le placement dont le rapport $r_i/p_i$ est maximum à condition que ce rapport soit positif. Dans le cas contraire (tous les $r_i$ étant négatifs ou nuls) la meilleure solution consiste à ne rien investir.

Géométriquement les solutions optimales correspondent aux sommets du polyèdre des solutions réalisables qui sont situés sur chaque axe et à l'origine (cf. fig. 1).

Optima de $Z^*(r)$

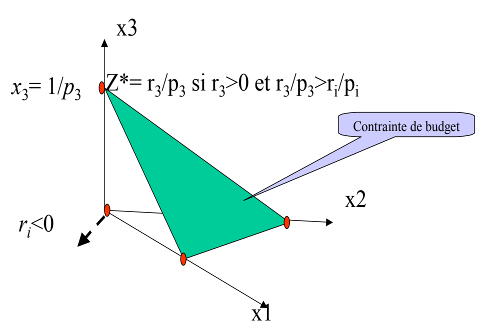



Figure 1. Points optimaux possibles en fonction des $r_i$

Les $n$+1 expressions de Z* : $Z_1^*$ , $Z_2^*$ ,… $Z_i^*$ ,… $Z^*_{n+1,}$ sont exprimées ci-dessous :

1. $Z_1^* = r_1/p_1$ lorsque $r_1 \geq 0$ et $r_1/p_1 \geq r_j/p_j$ quel que soit $j \neq 1$

…

i $Z_i^* = r_i/p_i$ lorsque $r_i \geq 0$ et $ri/pi \geq rj/pj$ quel que soit $j \neq i$

…

n+1 $Z^*_{n+1} = 0$ lorsque $r_i \leq 0$ pour tout $i$

L'hypercube des $r_i$ est partitionné par les domaines définis ci-dessus. Les classes de la partition ont des frontières communes lorsque $r_i/p_i = r_j/p_j$

Pour un portefeuille ($x_1$,..$x_n$ ) fixé il n'y a donc que $n$+1 expressions du regret en fonction des $n$+1 domaines de $r_i$ définis ci-dessus:

Domaine1 : $R_1 = r_1/p_1 - \sum_i r_i x_i = (1/p_1 - x_1)r_1 - \sum_{j \neq 1} r_j x_j$

Puisque $x_1 \leq 1/p_1$ (contrainte de budget), R1 prend sa valeur maximum

$R^*_1 = a/p_1 - ax_1 + sx_2 + .. + sx_n$ , obtenue pour $r_1 = a$ et $r_j = -s$ ; $j \neq 1$ qui est bien un élément du domaine $\{ r_i \mid r_1 \geq 0$ et $r_1/p_1 \geq r_j/p_j\ \forall j \}$. Dans le contexte, cela signifie que le regret est constitué du manque à gagner de ne pas avoir tout investi sur l'investissement 1 moins les « gains algébriques » obtenus sur les autres investissements. Le regret prend sa valeur maximale lorsque $r_1$ prend sa valeur maximale et que les autres rapports prennent leur valeur minimale ( la valeur du capital s'est partiellement réduite).

Domaine i : $R_i = (r_i/p_i) - \sum_i r_i x_i = (1/p_i - x_i)r_i - \sum_{j \neq i} r_j x_j$

qui prend sa valeur maximum $R^*_i = a/p_i - ax_i + \sum_{j \neq i} sx_j = a/p_i - (s+a)x_i + \sum_j sx_j$, obtenue pour $r_j = -s$ et $r_i = a$

Domaine n+1 : $R_{n+1} = 0 - (r_1 x_1 + .. + r_n x_n)$ dont la valeur maximale vaut $R^*_{n+1} = sx_1 + … + sx_n$ obtenue pour $r_1 = r_n = -s$

Par hypothèse, une solution robuste est définie par $\text{Min}_{x1x2..} \{\text{Max}_{\ ri} (R_1,.., R_i,.., R_{n+1})\}$

Or, Max $_{ri}$ ($R_1$,.., $R_i$,.., $R_{n+1}$)} = Max ($R_1$*,.., $R_i$*,.., $R_{n+1}$*) puisque les R*i majorent les Ri dans chaque domaine. En pratique il n'y a donc que $n$+1 « états de la nature » à prendre en compte et non pas une infinité. •

Recherchons les $R_i$* maximaux, $R_i$* = Max ($R_1$*,.., $R_i$*,.., $R^*_{n+1}$)

**Notation** : Dans la suite nous notons $M$ le plus petit entier supérieur ou égal à $1+a/s$.

**Théorème 1** : La solution robuste $x^* = (x^*_1, x^*_2, \ldots x^*_n)$ définie par l'optimisation : $\mathrm{Min}_{x1,..,xn}$ {Max ($R^*_1$,.., $R^*_i$,.., $R^*_n$)} est caractérisée par :

_ $x^*_{M+1} = x^*_{M+2} = \ldots = x^*_n = 0$ ; seules $x^*_1$ , $x^*_2$ , ...$x^*_M$ peuvent être non nulles .

_ $x^*_1$ est strictement positive.

**Preuve**:

La solution « robuste » peut être trouvée par le programme linéaire classique à $n$+1 variables et $n$ + 2 contraintes suivant :

Min λ ;

$R^*_i \leq \lambda$ pour tout i = 1 , ..$n$+1

$p_1 x_1 + p_2 x_{2+} \ldots + p_n x_{n,} \leq 1$

Soit compte tenu de l'expression des $R^*i$

P : Min λ

$\lambda + a\, x_1 - s x_2 - s\, x_3 \;\ldots\ldots -s\, x_n \;\geq a/p_1$ (P1)

$\lambda - s\, x_1 + a x_2 - s\, x_3 \;\ldots\ldots -s\, x_n \;\geq a/p_2$ (P2)

…..

$\lambda - s x_1 - s\, x_2 - s\, x_3 \;\ldots\ldots + a\, x_n \;\geq a/p_n$ (Pn)

$\lambda - s x_1 - s\, x_2 - s\, x_3 \;\ldots\ldots -s\, x_n \;\geq 0$ ($P_{n+1}$)

$0 - p_1 x_1 - p_2 x_2 \ldots - p_n x_{n,} \geq -1$

$x_1 ,\; x_2 ,\, x_3 \;\ldots\ldots, x_n \;\geq 0$

$\lambda$ de signe quelconque (en fait puisqu'il majore un regret il est toujours non négatif).

Le programme dual (en variables positives $u_1$, $u_2$, ..$u_n$, $u_{n+1}$, $v$) du programme précédent s'écrit :

D : Max $a\, u_1/p1 + a\, u_2/p_2 + \ldots + a\, u_n/p_n - v$

$u_1 + u_2 + \ldots + u_n + u_{n+1} = 1$ (D0)

$au_1 - su_2 - \ldots - su_n - su_{n+1} - p_1 v \leq 0$ ou $(s+a)\, u_1 \leq s + p_1 v$ (D1)

$-u_1 + a\, u_2 - \ldots - u_n - u_{n+1} - p_2 v \leq 0$ ou $(s+a)\, u_2 \leq s + p_2 v$ (D2)

$-su_1 - su_2 + a\, u_3 - \ldots - su_n - su_{n+1} - p_3 v \leq 0$ ou $(s+a)\, u_3 \leq s + p_3 v$ (D3)

…

$su_1 + su_2 + \ldots - au_n - su_{n+1} - p_n v \leq 0$ ou $(s+a)\, u_n \leq s + p_n v$ (Dn)

$u_1, u_2, \ldots, u_n, u_{n+1}, v \geq 0$

**Lemme 2 :** A l'optimum du Dual, il y a au maximum les $M$ premières variables non nulles.

**Preuve :**

Pour $v$ fixé, il faut donner la valeur maximale aux premières valeurs de $u_i$ puisque les coefficients de la fonction économique sont décroissants et que la borne supérieure des $u_i$ ne dépend que de $v$.

1. Cas $v^* = 0$

Les contraintes D1, D2, Di,…Dn s'écrivent : $u_i \leq s / (s+a)$

Compte tenu de D0, de la positivité de $a$ , et de la décroissance des $a / p_i$ , la solution du programme est dans ce cas :

$u_i = s / (s+a)$ pour $i \leq M-1$ ; $u_M = 1 - \frac{M-1}{s+a} s$ ; $u_i = 0$, pour $M+1 \leq i \leq n$ .

La valeur de la fonction économique est alors : $\frac{as}{s+a} \sum_{i=1}^{[M-1]} \frac{1}{p_i} + (1 - \frac{M-1}{(s+a)}) \frac{as}{p_M}$

Notons que si $M < n$ , les contraintes $D_i$ pour $M \leq i \leq n$ ne sont pas saturées à l'optimum

2. cas $v^* > 0$

Dans ce cas les seconds membres augmentent en fonction du rang i de chaque contrainte car les $p_i$ sont croissants. La solution construite avec les $u_i$ de la solution précédente (trouvée pour $v = 0$) et $v^*$ est une solution réalisable. Si cette solution n'est pas optimale, il existe une solution voisine qui est préférable (propriété classique liée à la convexité du domaine des solutions réalisables). Compte tenu de la contrainte D0 toute augmentation d'une variable $u_i$ pour $M \leq i \leq n$ diminue la valeur d'au moins une variable de base de la solution ($i < M$) et par suite pénalise le critère puisque les $1/p_i$ sont décroissant. Donc toute solution possédant une composante $u_i$ strictement positive pour $i > M$ ne peut êre optimale. Il y a donc au plus les $M$ premières variables non nulles et au plus les $M$ premières contraintes du dual sont saturées.

$$u^*_i = \frac{1}{s+a}(s+p_i v^*) \text{ pour } i < M\,;\; u^*_M = 1 - \sum_{i=1}^{M-1} u_i^*\,;\; u_i = 0, \text{ pour } M+1 \leq i \leq n \;\bullet$$

En conséquence d'après les conditions de complémentarité vérifiées pour les solutions optimales des deux programmes (primal et dual), il suit que :

$x^*_{M+1} = x^*_{M+2} = \ldots = x^*_n = 0$ puisque $D_{M+1}, \ldots, D_n$ ne sont pas saturées. Les investissements de rangs strictement supérieurs à $M$ ne sont jamais utilisés. •

**Lemme 3 :** A l'optimum du programme dual, si $u_i^* > 0$ alors $u_1^*, u_2^*, \ldots, u^*_{i-1} > 0$.

**Preuve** :

En effet, quelle que soit la valeur optimale $v^*$ les variables sont bornées par des valeurs strictement positives et par suite, selon le résultat classique du problème « du sac à dos » en variables continues, la décroissance des $a / p_i$ implique que les variables précédentes de $u_i$ ($u_1, u_2, \ldots, u_{i-1}$) doivent prendre leur valeurs maximales. •

**Lemme 4 :** A l'optimum on a toujours $u_1^*, u_2^* > 0$

**Preuve :**

$u_1^*$ ne peut pas être la seule valeurs non nulle car dans ce cas $u_1^* = 1$ d'après D0 et $s+a \leq s+p_1 v$ d'après D1, donc $v^* \geq a / p_1$. La fonction « objectif » du problème D est négative ou nulle ce qui n'est manifestement pas optimal ! D'après le Lemme 3 : on a donc $u_1^*, u_2^* > 0$ •

Il s'ensuit d'après les conditions de complémentarité entre programme primal et dual que les contraintes P1 et P2 sont saturées ce qui se traduit par $R^*_1(x^*) = R^*_2(x^*) = \lambda$. Ces deux regrets sont maximaux et les variables $x_1^*$ et $x_2^*$ sont liées par la relation :

$$x^*_1 - x^*_2 = \frac{a}{a+s}\left(\frac{1}{p_1} - \frac{1}{p_2}\right).$$

Ce qui implique comme conséquences :

_ L'investissement dans le placement 1 est toujours strictement positif.

_ Il est supérieur à l'investissement du placement 2 , cela se justifie par un coût plus faible pour 1 et une absence d'information sur les rentabilités.

_ L'écart entre ces investissements est inversement proportionnel à la volatilité ($a + s$) pour une rentabilité maximale ($a$) fixée. Cet écart croit avec la rentabilité maximale supposée et décroît avec le risque.

**Théorème 2 :** Un point optimal pour un jeu donné de $r_i$ (point extrême du polyèdre) ne peut être solution robuste.

**Preuve** :

En effet, ces points ont pour coordonnées (0,0,..,1/pi,0..0) . D'après la première conclusion du Lemme 3 : cela ne peut être que le point ($x_1$=1/$p_1$,0,0…,0). Pour ce point les regrets sont $R^*_1$=0 (évidemment puisqu'on est à l'optimum) ; $R_i^* = a/p_i - 0 + s\, x_1$ pour 1< i < n+1 ; $R^*_{n+1}= s\, x_1$. En diminuant $x_1$ d'une petite quantité on diminue les $R^*_i$ pour i>1 tout en maintenant $R^*_1$ inférieur aux $R^*_i$ pour i >1. Le critère du regret maximum est donc amélioré. •

Le choix d'une solution optimale pour une hypothèse de rentabilités fixées n'est donc pas optimal.

**Théorème 3 :** Max ($R^*_1$,.., $R^*_i$,.., $R^*_n$, $R^*_{n+1}$) = $R_1^*= R_2^*$,..,= $R^*_M$, c'est à dire que les regrets maximum ne sont à rechercher que sur les champs de paramètres 1,2 ... *M*

Ce résultat vient de la positivité de $u^*_i$ pour $i \le M$ *; au moins* les *M* premières contraintes du primal sont saturées et de valeur égale

# 3 Applications

## - Cas où $a \le s$, $n \ge 2$

Ce cas traduit que le risque est plus important que le gain, par exemple on peut gagner 30% du capital investi ($a$ = 0,3) et perdre 50 % ($s$ = 0,5).

Dans ce cas $M = 2 > 1+ a/s$

Par suite, seules $x_1$, $x_2$ sont positives ou nulle (pour $x_2$), pour i ≥3 $R^*_i = a/p_i + s\, x_1 + s\, x_2$ pour i≥3, *on a donc* $R^*_3 > R^*_4 > ... > R^*_n > R^*_{n+1} = x_1 + x_2$ (d'après les hypothèses de croissance des $p_i$ ). $R^*_4, R^*_5, .., R^*_{n+1}$ n'interviennent pas dans le calcul du maximum. En revanche $R^*_3$ peut être égale à $R^*1$ *et* $R^*2$.

Par suite, seules les contraintes *P1, P2 et P3* sont nécessaires à la résolution de *P*.

*L*es solutions « robustes » sont dans un domaine, ***R***, à une dimension défini par $R^*_1=R^*_2$ *et* $R^*_3 \leq (R^*_1+R^*_2)/2=R^*_1=R^*_2$ , soit

1. $x^*_1 - x^*_2 = a/(s+a)\times(1/p_1 - 1/p_2)$ *(c1)*
2. $x^*_1 + x^*_2 \leq a/(s+a)(1/p_1 + 1/p_2 - 2/p_3)$. *(c2)*

Dans ce domaine $R^*2$ (et $R^*1$) doivent être le plus petit possible.

Or $R^*_2 = a/p_2 + (s-a)\, x_2 + s\, x_1 - s x_2 = a/p_2 + (s-a)\, x_2 + s\, a/(s+a)\times(1/p_1 - 1/p_2)$ , $x_2$ doit donc être nul si $a < s$ ; alors $x_1 = a/(s+a)\times(1/p_1 - 1/p_2)$

Interprétation : si le risque est plus fort que le gain, le portefeuille n'est constitué que de l'actif 1 (le moins onéreux). Le budget consacré est : $a/(s+a)\times(1 - p_1/p_2)$. Il est d'autant plus élevé que l'écart entre $p_1$ et $p_2$ est grand.

Le budget n'est jamais consommé intégralement. En effet, *c1* et *c2* impliquent :

_ $x^*_1 \leq a/(s+a)(1/p_1 - 1/p_3)$.

_ $x^*_2 \leq a/(s+a)(1/p_2 - 1/p_3)$.

D'où $p_1 x^*_1 + p_2 x^*_2 \leq a/(s+a)(2 - p_1/p_3 - p_2/p_3) < 0{,}5 \times 2 = 1$.

Puisque l'écart $x_1 - x_2$ est constant le budget minimum est obtenu avec $x_2 = 0$, ce budget minimum est donc : $a/(s+a)\times(1 - p_1/p_2)$ .

Si $a = s$, $x_1$ et $x_2$ peuvent prendre n'importe quelles valeurs (respectant *c1* et *c2*)

### - Cas où *a* > *s*

Dans ce cas il y a en général min (*n*, *M-1*) ou min(*n*, *M*) investissements avec consommation totale ou partielle du budget. La diversification des investissements est liée au rapport a/s.

# 4 Conclusion

Il est amusant de remarquer que les conseils des gestionnaires de patrimoine préconisent de préférence une diversification des investissements plutôt que le choix d'un nombre réduit d'investissements au moindre coût d'entrée.

Ceci vient peut être du fait que pour le gestionnaire,

- D'une part, le critère implicite est vraisemblablement l'espérance mathématique du gain et non pas le risque encouru dans le pire cas.

- D'autre part les risques ou les gains des différents investissements sont estimés différemment selon le type de placement.

Notre analyse qui ne fait aucune distinction entre les investissements en termes d'informations, hormis le coût d'entrée, s'appliquerait donc avec plus de pertinence à un groupe homogène -en termes de volatilité- de placements.

Les résultats obtenus traduisent le caractère pessimiste du critère de Savage qui conduit généralement à un choix très prudent.

Il serait intéressant d'introduire dans cette analyse une diversification des taux de gain ou de perte en fonction du type d'investissement en conservant des relations du type $a$-$s$ = constante ou $a/s$ = constante traduisant l'équilibre entre espérance de gain et risques de pertes afin de voir quelles conclusions subsistent.

L'optimisation des gains nets ($r_i$-$c_i$) ne modifie pas les conclusions obtenues d'après les calculs numériques effectués, la démonstration légèrement plus complexe reste à être effectuée.

# Références

Averbakh, Minmax regret linear resource allocation problems, Operations Research Letters 32 (2004), p 174-180

P. Kouvelis, G. Yu, Robust Discrete Optimization and its Applications, Kluwer Academic Publishers, Boston, 1997.