

CAHIER DU LAMSADE

192

Mars 2002

Réflexion autour de nouvelles notions pour l'analyse des
algorithmes d'approximation

M. Demange, V. Th. Paschos

Table des matières

Abstract	iii
Résumé	iv
1 Introduction	1
2 Notions de base – le formalisme usuel	2
2.1 Le cadre	2
2.2 Généralités sur les algorithmes d'approximation	4
2.2.1 La classe NPO	4
2.2.2 Algorithmes approchés et mesures d'approximation	4
2.2.3 Résultats d'approximation	6
2.3 Enjeux de l'approximation à garanties de performances et classification des concepts	6
2.4 Motivations pour enrichir le formalisme de l'approximation polynomiale	7
3 Approximabilité – une première classification de NPO	8
3.1 Un exemple : une brève histoire de l'approximation du problème de stable maximum	9
3.2 Algorithmes et chaînes d'approximation	10
3.3 Niveaux d'approximation	11
4 Notions de limites	12
4.1 Convergence d'une chaîne d'approximation polynomiale	12
4.2 Notions asymptotiques	12
4.2.1 Définition usuelle	13
4.2.2 Commentaires	13
4.2.3 Ordre de difficulté	18
4.2.4 Proposition de définition	19
4.3 Convergence asymptotique	20
5 Exemples	20
5.1 Algorithmes d'approximation	21
5.1.1 Classe APX	21
5.1.2 Rapports dépendant de l'instance	21
5.1.3 Rapports fonction d'autres paramètres	22
5.2 Chaînes d'approximation	24
5.3 Approximation différentielle	25
6 Comparer les problèmes : un complément pour structurer NPO	26
6.1 Motivations	26
6.2 Généalogie	27
6.3 Réduction à expansion fonctionnelle FP	29
6.4 Réductions du troisième type	30
6.5 Quelques exemples	31
6.5.1 Quelques grands classiques	31
6.5.2 Stratégie maître-esclave	31
6.5.3 Liens entre versions pondérée et non-pondérée	31

6.5.4	Sous graphe induit k -colorable	33
6.5.5	Du stable à la clique	34
6.5.6	Réductions entre les cadres classique et différentiel	35
7	Difficulté des instances d'un problème	35
7.1	Deux notions de simplicité	36
7.2	Problèmes radiaux	37
7.2.1	Problèmes héréditaires	37
7.2.2	Une généralisation	38
7.2.3	Liens avec les notions de simplicité	39
7.3	Structuration par élargissement d'une classe	40
7.3.1	Stabilité d'un algorithme par rapport à une fonction d'éloignement	41
7.3.2	Autour de la propriété de KÖNIG-EGERVARY	42
7.4	Réduction à des sous-problèmes : autres exemples concernant le stable	44
7.4.1	Stable et rapports $O(n^{\epsilon-1})$	44
7.4.2	La classe STABLE_k	45
7.4.3	La classe de NEMHAUSER et TROTTER	46
7.5	Famille critique d'instances	46
8	Conclusion	47
	Références	50
A	Quelques problèmes NPO	53
B	Rappel de quelques définitions de base	55

Thoughts about new notions for the analysis of approximation algorithms

Abstract

The main objective of the polynomial approximation is the development of polynomial time algorithms for **NP**-hard problems, these algorithms guaranteeing feasible solutions lying “as near as possible” to the optimal ones. We first introduce the key-concepts of the polynomial approximation and then we present the main lines of a new formalism. Our purposes are, on the one hand, to present this theory and its objectives and, on the other hand, to discuss the appropriateness and the pertinence of its constitutive elements, as people knew them until now, and to propose their enrichment. The different stages of this work are accompanied by several illustrative examples. So, this paper is addressed to both domain researchers and non-specialist readers. We particularly quote the great interest, both theoretical and operational, to construct an internal structure for the class **NPO** (of the optimization problems in **NP**). For this reason we devise the notions presented in two categories. The tools of the former allow the individual evaluation of the approximability properties of any **NP**-hard problem. We present and discuss notions as algorithmic chain, approximation level, or even notions of limits (with respect to algorithmic chains, or to problems instances). The tools of the second category allow the comparison between different problems regarding their respective approximability properties. We find here the central complexity object, the reduction (adapted, of course, in the framework of the polynomial approximation). We propose a new definition unifying the several approximation preserving reductions known in the literature and also extending their scope. We try to justify the interest of this new definition by several speaking examples. The last part of the paper applies the different concepts discussed formerly in the study of the hardness (approximation intractability) of an instance and in the apprehension of the structure of the set of hard instances of a problem. Then, many distinct situations appear, under our point of view, as complementary ones. This part of the paper allows us to also draw directions for further research.

Keywords: **NP**, **NP**-complete, complexity, polynomial approximation.

Réflexion autour de nouvelles notions pour l'analyse des algorithmes d'approximation

Résumé

Nous présentons les principales caractéristiques d'un nouveau formalisme pour l'approximation polynomiale (algorithmique polynomiale à garanties de performances pour les problèmes **NP**-difficiles). Cette présentation est l'occasion d'un regard critique sur ce domaine et de discussions sur la pertinence des notions usuelles. Elle est aussi l'occasion de se familiariser avec l'approximation polynomiale et de comprendre ses enjeux et ses méthodes. Cet article s'adresse donc autant aux spécialistes qu'aux non spécialistes de ce domaine. Les différentes étapes de ce travail sont illustrées par de nombreux exemples, choisis pour leur pertinence et leur simplicité. Nous insistons en particulier sur l'intérêt, tant théorique qu'opérationnel, de mettre en évidence une structure au sein de la classe **NPO** des problèmes d'optimisation de **NP**. Ainsi, nous classons les notions d'approximation polynomiale en deux catégories. Les outils de la première permettent d'évaluer, dans l'absolu, les propriétés d'approximation de problèmes difficiles; nous discutons notamment les notions de chaînes d'approximation, de niveau d'approximation ainsi que deux notions de limites (par rapport à une suite d'algorithmes et par rapport aux instances). Les outils de la seconde catégorie permettent de comparer, du point de vue de l'approximation, différents problèmes : il s'agit de la notion, très fructueuse, de réduction en approximation. Nous proposons une définition qui unifie, sous le point de vue de l'approximation, les réductions existantes. Des exemples variés mettent en évidence l'intérêt et les enjeux de cette notion. Enfin, nous appliquons les différents concepts pour étudier la difficulté des instances d'un problème et la structure de leur ensemble. De nombreuses situations différentes apparaissent ainsi comme complémentaires. Cette partie nous permet également de montrer de nouvelles directions d'études.

Mots-clé : **NP**, **NP**-complet, complexité, approximation polynomiale.

1 Introduction

L'approximation polynomiale a pour objet la conception et l'analyse d'algorithmes polynomiaux avec garanties de performances pour des problèmes difficiles n'admettant pas d'algorithme exact.

Nous proposons, dans cet article, une promenade au sein d'un nouveau formalisme pour l'algorithmique à garanties de performances. Elle est l'occasion de découvrir certains aspects de cette théorie et d'engager une réflexion sur ses enjeux et sur la pertinence de ses concepts. La démarche et les raisons qui nous ont conduit à proposer ce formalisme nous semblent intéressants, non seulement pour tout spécialiste du domaine, mais aussi pour quiconque désirant en avoir un bref aperçu. Aux premiers nous proposons les rudiments d'un débat auquel nous les invitons à se joindre. Ce débat nous paraît important pour le développement cohérent, la bonne utilisation et la promotion de l'approximation à garanties de performances. Aux seconds nous voulons montrer un aspect souvent ignoré de ce champs de recherche qui, comme beaucoup d'autres, se montre souvent sous ses traits les plus techniques réservant aux spécialistes l'élaboration des concepts et les travaux sur ses fondements. Pourtant, ces aspects nous semblent au moins aussi importants que les techniques de démonstration pour aborder ce domaine et comprendre ses enjeux. C'est pourquoi nous invitons l'œil extérieur à partager ces premiers éléments de réflexion. Les concepts décrits dans cet article sont accessibles sans connaissance préalable particulière et sont emprunts d'une certaine vision de l'algorithmique à garanties de performances. Notre souci a en effet été, dans leur conception, de mieux répondre aux besoins et aux enjeux de l'approximation polynomiale ; ils en deviennent ainsi une bonne illustration. Enfin, la démarche vers ce formalisme, ses justifications et les besoins auxquels elle essaie de répondre, ont leur intérêt, même placés hors du contexte de l'approximation. En effet, comme nous le montrons dans la suite, l'approximation à garanties de performances est à un point de son développement où un besoin conceptuel se fait sentir. Des situations analogues peuvent se présenter dans moult domaines scientifiques et il serait intéressant de comparer l'expérience que nous relatons ici à des travaux du même genre dans d'autres thématiques.

Notre objet n'est pas de développer les détails de ce formalisme mais plutôt d'en retirer la logique générale, sa pertinence, ses conséquences pratiques et les raisons qui nous ont conduit à le concevoir. Dans cette optique, les résultats d'approximation cités dans ce document ont valeur d'exemples. Certains ont été spécialement conçus pour illustrer notre propos ; d'autres, par contre, existent dans la littérature mais sont repris ici (parfois même seulement mentionnés dans le texte) pour motiver notre démarche. Pour ces derniers, seules les preuves les plus accessibles illustrant un point particulier sont reprises. Par souci de lisibilité et de continuité du document, nous avons sélectionné deux problèmes – *le stable maximum* (noté STABLE dans ce qui suit) et le *bin-packing* (BINPACKING) – auxquels se rapportent la majorité de nos exemples. D'autres problèmes, notamment *la clique maximum* (noté CLIQUE dans ce qui suit), *la coloration minimum* (COLORATION) et *la couverture d'ensembles* (noté H-TRANSVERSAL) sont mentionnés ponctuellement pour des exemples significatifs n'ayant pas d'équivalent pour nos deux problèmes de référence. Tous les problèmes cités dans le document sont définis en annexe dans la section A.

Nous rappelons, dans la section 2, les notions de base sur lesquelles repose la suite de ce travail. Nous situons l'approximation polynomiale par rapport à la théorie de la complexité et aux différentes méthodes de résolution de problèmes difficiles (section 2.1) et détaillons ses enjeux (section 2.3). Nous présentons notamment (section 2.2) les notions de problème d'optimisation, d'algorithme approché et de rapport d'approximation. Dans le paragraphe 2.4 nous proposons alors une synthèse des principales motivations qui ont conduit à compléter ce formalisme usuel.

Dans la section 3 nous discutons de premières extensions des notions usuelles permettant d'exprimer les possibilités de résolution efficace d'un problème. Un petit historique concernant

l'approximation du problème STABLE (paragraphe 3.1) motive ces extensions. Nous présentons alors les notions de *chaînes d'approximation* (paragraphe 3.2) et de *niveau d'approximation* (paragraphe 3.3). Ces concepts ouvrent la voie à différentes notions de limites présentées dans la section 4. En particulier, nous discutons la définition de *rapport asymptotique* (section 4.2) et développons, à cet effet, plusieurs exemples. Le concept d'*ordre de difficulté* (paragraphe 4.2.3) est central pour cette question. La section 5 regroupe des exemples de synthèse qui illustrent ces notions et les motivent.

Après le développement de concepts permettant d'évaluer directement les possibilités d'approximation de problèmes, nous envisageons, dans la section 6, la notion de *réduction en approximation* qui permet de comparer les possibilités de résolution efficace de problèmes différents. Nous présentons, dans le paragraphe 6.2, l'évolution des différents types de réductions afin de motiver l'intérêt de les unifier (paragraphe 6.3). Nous donnons alors quelques exemples de réductions mettant en évidence les possibilités de ce concept et illustrant le type de questions qu'il permet d'étudier. Enfin, dans la partie 7, nous envisageons la question de la structure des instances d'un problème. Nous montrons comment les outils de l'approximation, et en particulier la notion de réduction, peuvent servir à étudier chaque problème. Cette section nous permet de montrer comment l'étude d'un problème et de la structure de ses instances est intimement liée à l'étude de la structure de l'ensemble des problèmes. Tant du point de vue opérationnel que structurel, les travaux sur chaque problème et ceux portant sur leur ensemble apparaissent comme complémentaires et similaires. Enfin, nous proposons en annexe (section A) une liste de problèmes d'optimisation utilisés dans ce document.

2 Notions de base – le formalisme usuel

2.1 Le cadre

La classe **NP** désigne l'ensemble des problèmes résolubles par une machine de TURING polynomiale non déterministe; il s'agit, au moins dans la théorie classique de la complexité, de l'univers de travail. La classe **P** (des *problèmes polynomiaux*) est l'ensemble des problèmes de **NP** résolus par une machine de TURING polynomiale déterministe. Dans ce type d'étude, la complexité algorithmique polynomiale correspond à ce qui est communément considéré comme une *bonne résolution*. À l'opposé, la classe **NP** contient les problèmes **NP-complets** (classe **NP-C**) qui sont, dans un certain sens, les problèmes les plus difficiles de **NP** : la résolution de l'un d'entre eux par une machine de TURING polynomiale déterministe permettrait la résolution de n'importe quel problème de **NP** par une machine de ce type. Parmi les problèmes connus et entrant dans le cadre de nombreux modèles industriels ou en sciences humaines, relativement peu sont polynomiaux: les principaux problèmes de **P** sont *les problèmes de tris, de parcours d'arbres et de graphes, certains problèmes de flots, d'affectation, des problèmes de cheminements optimaux (avec notamment, comme application, les problèmes d'ordonnancement les plus simples et, plus généralement, l'optimisation séquentielle), le problème de couplage maximum d'un graphe et la programmation linéaire en variables continues*. Ces problèmes suscitent un intérêt particulier, non seulement parce qu'ils ont de nombreuses applications, mais aussi parce qu'ils constituent les outils de base de l'algorithmique polynomiale. Si les problèmes polynomiaux sont relativement rares, de très nombreux modèles réels conduisent par contre à des problèmes **NP-complets**. Parmi eux, citons bien sûr le fameux problème de *satisfiabilité* qui fut le premier problème **NP-complet** à être mis en évidence par COOK ([12]). Mentionnons aussi *le problème du voyageur de commerce* (TSP), STABLE, *l'arbre de STEINER*, COLORATION, les principaux problèmes de partage, de découpe et de couverture minimale, la plupart des problèmes d'ordonnancement, *le problème du sac à dos* (KS) et, plus généralement, *la programmation linéaire en variables entières*. La découverte, depuis les travaux de COOK et de KARP ([12, 43]), d'une multitude de

problèmes **NP**-complets issus d'applications réelles a mis en évidence les limites de la résolution informatique de problèmes concrets. Mais surtout, ce constat justifie les efforts actuels en vue de résoudre, au moins partiellement, de tels problèmes de manière efficace.

Dans cette optique, plusieurs directions de recherches ont été développées. Parmi celles-ci, mentionnons d'abord l'identification de cas particuliers (sous-problèmes) polynomiaux d'un problème général difficile et la mise au point de méthodes exactes. La première est particulièrement intéressante pour comprendre ce qui fait la difficulté d'un problème et pour mettre en évidence des cas simples qui interviennent parfois de manière naturelle dans les modèles généraux. Les méthodes exactes, quant à elles, permettent de traiter des instances de taille modérée (ce terme prenant des significations très différentes d'un modèle à un autre); elles sont fondées sur une recherche exhaustive arborescente pour laquelle on réussit à s'affranchir, par des arguments de bornes, d'une partie de l'espace de recherche.

Un autre point de vue pour résoudre efficacement des problèmes intrinsèquement difficiles est d'envisager des méthodes approchées. L'idée sous-jacente est qu'une bonne solution, éventuellement non optimale mais obtenue en temps raisonnable, vaut parfois mieux qu'une solution optimale nécessitant un temps de calcul trop élevé. D'ailleurs, lorsque la taille de l'instance ne permet pas sa résolution complète ou lorsque les sous-problèmes identifiés comme faciles n'ont aucune pertinence du point de vue des besoins, les deux premiers points de vue sont inopérants. L'idée de gagner en temps de calcul (ou de rendre le calcul possible) au prix de la qualité de la solution peut alors séduire, voire même s'imposer. Là encore, cette issue n'est envisageable que si le problème s'y prête. En effet, dans certains cas une solution non exacte n'est pas exploitable et parfois n'a même pas de sens. C'est en particulier le cas pour un problème de décision dont la réponse ne peut être qu'alternative. Les méthodes approchées ne sont envisagées que pour des problèmes d'optimisation pour lesquels on considère qu'une solution satisfaisant les contraintes peut être exploitée sachant qu'elle le sera d'autant mieux que sa valeur objective est bonne. Les contraintes du problème représentent ce que doit absolument vérifier une solution, l'objectif détermine sa qualité. Les méthodes approchées diffèrent les unes des autres par le type de compromis qualité/complexité qu'elles offrent. Nous en distinguons deux classes: les heuristiques et les algorithmes polynomiaux à garanties de performances. Les premières, très utilisées en pratique, n'offrent aucune garantie a priori. Il s'agit de stratégies, de types très variés, dont le principe repose sur des arguments intuitifs, voire sur des paradigmes de phénomènes naturels mais qui ne sont justifiées que par les résultats qu'elles offrent. Aussi, le recours à des plans d'expérience est souvent retenu pour tester des heuristiques. Dans certains cas, ces méthodes donnent lieu à des analyses complétant les observations; elles sont systématiques (et même imposées dans les définitions) dans le cadre de l'approximation polynomiale. En effet, ce domaine se définit par des règles strictes que doivent satisfaire les algorithmes qu'il étudie. Le principe est de se limiter à l'emploi de *méthodes polynomiales* en exigeant de plus des *garanties absolues* sur la qualité des solutions obtenues. Jusqu'à présent, ce choix quant aux garanties de complexité et de qualité des solutions obtenues distingue l'approximation polynomiale des autres méthodes approchées; mais il serait tout à fait envisageable et particulièrement intéressant d'étudier le comportement des problèmes difficiles par rapport à un autre compromis complexité/qualité que le couple *polynomial/garanties absolues*.

D'un point de vue opérationnel, la pertinence de ces approches dépend des besoins spécifiques de chaque application. Dans la pratique, le recours à des méthodes heuristiques reste majoritaire, ce qui peut être attribué non seulement à leur bon comportement en pratique mais aussi à une relative méconnaissance des particularités de l'approximation polynomiale. L'autre raison est le pessimisme de certains résultats d'approximation qui provient essentiellement des restrictions que ce cadre impose. Soulignons néanmoins que les méthodes polynomiales peuvent avoir, dans la pratique, des comportements bien meilleurs que les bornes qu'elles garantissent. D'autant plus

qu'en combinant une heuristique et un algorithme polynomial à garanties de performances, on peut bénéficier du bon comportement en pratique de la première et des garanties d'approximation du second.

2.2 Généralités sur les algorithmes d'approximation

2.2.1 La classe NPO

L'approximation polynomiale restreint son champs d'étude à certains problèmes pouvant s'exprimer en termes d'optimisation. Les problèmes **NP** sont définis à l'origine ([28]) comme des problèmes de décision exprimés sous forme d'une question ; la *résolution* du problème consiste alors à apporter une réponse (*oui* ou *non*). Pour de tels problèmes, il est difficile de concevoir une notion de résolution approchée. Par contre, pour certains d'entre eux, il est possible d'associer une *version optimisation* exprimée par un programme mathématique (optimisation d'une fonction réelle sous contraintes). À chaque instance on associe une question portant sur l'existence ou non d'une solution meilleure qu'un seuil fixé. C'est par ce biais qu'on peut faire le lien entre les problèmes d'optimisation et les problèmes de décision de la classe **NP**. Plus précisément, nous rappelons ici la définition formelle des problèmes d'optimisation.

Définition 1. Un problème élémentaire ou « instance » (notée I) est un programme mathématique de la forme

$$I : \begin{cases} \text{opt} & v(\vec{x}) \\ & \vec{x} \in \mathcal{C} \\ & x_i \in \{0, 1\} \end{cases}$$

où $\mathcal{C}$ désigne ce qui est communément appelé « contraintes ». Un « problème d'optimisation combinatoire **NP** (resp., **NP-complet**) » est un ensemble d'instances tel que l'ensemble correspondant des questions « pour un M donné, existe-t-il $\vec{x} \in \mathcal{C}$ tel que $v(\vec{x})\theta M$? » (où θ vaut $\geq$ (resp., $\leq$) si opt vaut max (resp., min)) est exactement l'ensemble des instances d'un problème **NP** (resp., **NP-complet**) au sens usuel de la théorie des langages ([28, 45]). L'ensemble des problème d'optimisation **NP** est noté **NPO**. ■

Un problème **NPO** est au moins aussi difficile que sa version décision puisque pour résoudre cette dernière il suffit de connaître la valeur optimale du problème d'optimisation. Lorsque sa version décision est **NP-complète**, le problème **NPO**, au moins aussi difficile qu'un problème **NP-complet**, est dit **NP-difficile**. Parmi les problèmes **NPO** qui sont **NP-difficiles**, citons **STABLE**, **COLORATION**, **TSP**, ou encore la programmation linéaire en variables entières.

Le principal intérêt d'exprimer un problème en termes d'optimisation est d'avoir à sa disposition, non seulement une notion de solution acceptable¹ (ou réalisable) du problème, mais aussi une notion de qualité d'une solution réalisable par référence à la valeur de l'objectif.

2.2.2 Algorithmes approchés et mesures d'approximation

Pour un tel problème, on appelle *algorithme approché* un algorithme polynomial par rapport à la taille des instances fournissant, pour toute instance I , une solution réalisable de valeur $\lambda(I)$. Un résultat d'approximation permet de caractériser la qualité de la solution approchée à l'aide d'une *mesure d'approximation* définie pour chaque instance. La mesure la plus utilisée est le rapport

$$\gamma(I) = \min \left\{ \frac{\lambda(I)}{\beta(I)}, \frac{\beta(I)}{\lambda(I)} \right\}$$

1. Cette notion comprend, par le biais des contraintes, ce qu'on désire imposer à n'importe quelle solution envisageable.

où $\beta(I)$ désigne la valeur optimale de l'instance I . D'autres mesures d'approximation peuvent être employées, notamment le *rapport différentiel* que nous avons défini, justifié et discuté dans [17, 20, 21]. Pour une instance I , il s'exprime par

$$\delta(I) = \frac{|\lambda(I) - \omega(I)|}{|\beta(I) - \omega(I)|}$$

où $\omega(I)$ désigne la *pire valeur* de l'instance I ; $\delta(I)$ mesure donc la position de la valeur garantie par l'algorithme dans l'intervalle entre la pire valeur et la meilleure.

Revenons en quelques lignes sur les raisons qui nous ont amené à définir ce rapport. Il s'agissait de répondre à quelques « paradoxes » connus du cadre classique qui sont directement imputables à la mesure utilisée. Tous reviennent à la remarque suivante : du point de vue du rapport classique, des transformations très naturelles et intuitivement anodines de problèmes **NPO** changent radicalement leur comportement du point de vue de l'approximation. Un exemple significatif de telles transformations est l'ajout d'une constante à la fonction objectif ; ceci ne change rien du point de vue de l'optimisation et peut pourtant modifier le comportement du problème si on considère le rapport d'approximation classique.

Examinons l'exemple de BINPACKING qui consiste à ranger une liste nombres positifs inférieurs à 1 en un nombre minimum de boîtes (appelées *bin*) de capacité totale 1 (la somme des nombres placés dans chaque boîte ne doit pas excéder 1). Ajouter à une instance un nombre k fixé de « 1 » ne modifie intuitivement pas l'instance puisque chacun de ces « 1 » ajoutés devra, dans toute solution, utiliser une boîte à lui tout seul. Ceci revient à ajouter k à la valeur objective de toute solution réalisable de l'instance initiale. Cette transformation, quasiment insignifiante, a pourtant une conséquence importante sur le rapport d'approximation classique : il suffit d'augmenter à souhait k pour rendre toute solution « bonne » (rapport proche de 1) pour le rapport classique. Un autre exemple célèbre est le lien entre le problème STABLE et la couverture de sommets minimum dans un graphe (noté G-TRANSVERSAL dans ce qui suit). Un stable d'un graphe est un ensemble de sommets deux à deux non liés par une arête ; une couverture de sommets est un ensemble de sommets qui « touche » toutes les arêtes. Il est très facile de se rendre compte que le complémentaire de toute couverture de sommets est un stable et réciproquement. Ainsi, maximiser le cardinal d'un ensemble stable devrait exactement revenir à minimiser celui du complémentaire d'un stable, *i.e.*, d'une couverture de sommets. Pourtant, du point de vue de l'approximation classique (rapport γ), le premier de ces problèmes est réputé très difficile à approximer² alors que le second admet des algorithmes offrant des garanties communément jugées comme bonnes. Contrairement au cas précédent, les valeurs des solutions ne subissent pas seulement une translation au cours de la transformation, mais une transformation affine tout aussi naturelle. Dans l'exemple choisi, la transformation affine ($x \mapsto n - x$ où n désigne le nombre de sommets) est décroissante, ce qui explique la correspondance entre un problème de maximisation et un problème de minimisation. Sur la base d'exemples de ce type, une réflexion sur la mesure d'approximation à utiliser a été menée dans [17, 20, 21]. Le rapport différentiel a alors été défini par une approche axiomatique imposant notamment la stabilité du rapport vis à vis de transformations affines de l'objectif. En outre, cette démarche nous a conduit à réfléchir à la notion de pire valeur qui s'est avérée particulièrement riche. Aujourd'hui, l'approximation de plusieurs problèmes **NPO** a été étudiée sous les rapports classique et différentiel. Les résultats montrent la complémentarité de ces mesures et l'intérêt qu'il y a à mener de front des études sous chacune d'elles. Précisons que non seulement les résultats d'approximation mais aussi les méthodes déployées diffèrent en fonction du rapport. L'un des points forts de cette étude a été d'engager une réflexion sur le concept de rapport d'approximation et de montrer ce qu'un tel

2. Nous nous autorisons ce néologisme pour désigner la démarche de l'approximation ; de même, l'adjectif *approximable* désigne la faculté d'un problème à être approximé.

débat pouvait apporter. Les réflexions sur d'autres concepts que nous relatons ici se situent dans la continuité logique de cette étude.

2.2.3 Résultats d'approximation

Étant donné une mesure d'approximation³ $\rho(I)$, un résultat d'approximation s'énonce comme une borne valable pour toute instance I . Comme pour la plupart des travaux de théorie de la complexité, on mesure donc le comportement d'un algorithme par rapport au pire cas, ce qui permet d'obtenir des garanties absolues valables pour toutes les instances.

Les résultats d'approximation mettent en évidence des différences de difficulté entre les problèmes **NP**-difficiles alors que leurs versions décision sont équivalentes par rapport aux réductions polynomiales. Certains problèmes **NP**-difficiles sont très bien approchés par un algorithme polynomial; Ks est l'exemple le plus favorable puisqu'il admet un *schéma complet d'approximation polynomial* ([39]), c'est-à-dire, une famille (indicée par $\epsilon \in]0, 1[$) d'algorithmes polynomiaux (à la fois en n et en $1/\epsilon$) A_ϵ telle que, pour tout ϵ , l'algorithme A_ϵ garantit un rapport γ supérieur à $1 - \epsilon$. Pour d'autres problèmes au contraire, on sait que de tels résultats ne sont pas possibles sous une hypothèse de complexité. Par exemple, pour STABLE, si $\mathbf{P} \neq \mathbf{NP}$, aucun algorithme polynomial ne peut garantir un rapport γ borné inférieurement par une constante $c > 0$ ([3]). Les *résultats positifs* correspondent à la conception d'un algorithme polynomial dont l'analyse met en évidence un certain niveau d'approximation. Les *résultats négatifs* correspondent à un seuil de qualité qu'on ne peut atteindre, pour un problème, sous une hypothèse fortement improbable du type $\mathbf{P} \neq \mathbf{NP}$. L'association des deux permet d'établir une hiérarchie des problèmes d'optimisation **NP**-complets par rapport à l'approximation. De ce point de vue, l'approximation polynomiale apparaît comme un outil fondamental pour la compréhension de la classe **NPO** et pour l'étude de sa structure.

L'un des aspects de cette théorie est donc l'étude de la difficulté intrinsèque des problèmes en offrant, par rapport à la théorie de la complexité, plusieurs niveaux de difficulté. La définition de la **NP**-complétude offre deux niveaux de difficulté, les classes **P** et **NP-C**. Par contre, en approximation, l'ensemble des problèmes d'optimisation **NP**-complets se décompose en couches d'équi-approximabilité.

2.3 Enjeux de l'approximation à garanties de performances et classification des concepts

Les types de résultats obtenus et leur grande diversité mettent en évidence deux enjeux complémentaires de l'approximation à garanties de performances qui distinguent ce domaine de celui des heuristiques. Le plus évident est la résolution efficace de problèmes difficiles, l'efficacité correspondant ici au caractère polynomial des algorithmes et aux garanties de qualité de la solution qu'ils offrent. Comme nous l'avons déjà signalé, cet *intérêt opérationnel* de l'approximation complète celui des heuristiques pour la résolution de problèmes réels. Mais l'approximation polynomiale se distingue par la *possibilité d'étudier une structure de la classe des problèmes difficiles fondée sur les propriétés d'approximation*. C'est son second enjeu qui s'avère particulièrement important. Il permet de mieux connaître la classe des problèmes difficiles avec pour principal objectif une meilleure compréhension du phénomène de difficulté, donc sa meilleure gestion. C'est en ce sens que les deux enjeux se rejoignent : aujourd'hui, les études structurelles et la mise en évidence de liens entre différents problèmes est à la base de nombreux résultats.

Chaque type de notion permet d'y contribuer. La description absolue des propriétés d'approximation d'un problème est naturelle du point de vue opérationnel pour rendre compte de

3. Ici, nous prendrons la convention d'une mesure dans l'intervalle $[0, 1]$ avec la valeur 1 pour la résolution exacte.

la qualité d'algorithmes d'approximation et des améliorations qu'ils apportent. Ce point de vue permet également de définir une hiérarchie des problèmes selon les possibilités de résolution, mais les classes ainsi obtenues n'ont qu'un intérêt descriptif qui est déjà très intéressant. Le second type permet de mettre en évidence des liens entre problèmes indépendamment des résultats d'approximation connus. Du point de vue structurel, l'intérêt est évident : il permet de définir des classes plus consistantes qui reposent sur des propriétés propres des problèmes. Ces notions ont donc une valeur explicative des comportements d'approximation en ce sens qu'elles aident à comprendre ce qui fait la difficulté d'un problème ou ce qui fait la différence entre plusieurs problèmes. Mais elles ont également un enjeu opérationnel très important puisqu'elles permettent de transférer à de nouveaux problèmes des résultats obtenus pour d'autres ; comme nous le verrons, de tels liens sont à la base de nombreux résultats nouveaux.

2.4 Motivations pour enrichir le formalisme de l'approximation polynomiale

Notre travail visant à enrichir le formalisme classique de l'approximation polynomiale se situe dans la continuité de la réflexion que nous avons engagée - et qui reste d'actualité - sur le rapport d'approximation.

Avant d'en aborder d'autres aspects, nous exposons dans cette section nos principales motivations à engager ce travail. Dans la continuité directe de la théorie de la complexité ([28]), les développements de l'approximation à garanties de performances ont surtout consisté, dans un premier temps, à établir un fond de résultats sur la base de quelques définitions fondatrices sans que se fasse sentir le besoin de les modifier ou de les étendre. Aujourd'hui un tel besoin apparaît aux yeux de nombreux chercheurs si l'on en juge par le nombre d'articles récents présentant des variantes de définitions en vue d'intégrer de nouveaux résultats. Le besoin d'unifier des définitions aux variantes multiples et d'augmenter l'expressivité du formalisme pour intégrer de manière satisfaisante des résultats de plus en plus fins ou pour envisager des problématiques nouvelles, la volonté de simplifier l'expression des résultats pour mieux en rendre compte et enfin l'intérêt d'engager une réflexion sur la pertinence des notions et leur adéquation avec les objectifs de l'approximation sont les maîtres mots de nos motivations. Nous en donnons ici quelques exemples.

L'explosion du nombre de définitions et le degré de technicité auquel elles aboutissent motive la volonté de les unifier, nous l'illustrerons en particulier par l'exemple des réductions en approximation (section 6). L'unification des notions permet non seulement de les simplifier, mais aussi de mieux les comparer. Ce travail permet aussi de répondre aux besoins actuels du domaine et d'envisager des problématiques qui ne trouvent pas leur place dans le formalisme usuel. En effet, beaucoup de concepts d'approximation polynomiale ont été conçus, à l'origine, dans un cadre relativement restrictif correspondant aux besoins et aux résultats de l'époque. Un exemple significatif est la définition originelle ([28]) du rapport d'approximation ; avec les notations que nous avons adoptées, il était défini comme la borne inférieure de $\gamma(I)$ pour I décrivant l'ensemble des instances. Cette définition est adaptée au cas d'un rapport d'approximation indépendant de l'instance mais devient totalement inopérante pour des rapports d'approximation non minorés pour lesquels l'infimum est nul. Or le besoin de prendre en compte des rapports dépendant de l'instance s'est fait très vite sentir puisque pour certains problèmes tels que STABLE, TSP, COLORATION, etc., un rapport d'approximation constant ne peut être garanti⁴ par un algorithme polynomial. Cependant, la simple adaptation des définitions (ce qui a souvent été utilisé) n'est, selon nous, pas complètement satisfaisante. En effet, le cas des rapports dépendant de l'instance soulève des problèmes spécifiques tels que le type de paramètres à prendre en compte dans le rapport d'approximation et la manière dont évolue la difficulté des instances en fonction de ces

4. Sous une hypothèse de complexité.

paramètres. L'exemple de STABLE, développé au paragraphe 3.1, illustre tout à fait cette question. Deux familles de résultats d'approximation ont été établies en parallèle : ceux exprimés en fonction du nombre de sommets du graphe et ceux exprimés en fonction de son degré maximum. Les formes de rapports d'approximation mis en jeu pour chaque paramètre sont très différentes. La question de savoir si tout rapport exprimé en fonction de l'un des paramètres peut être transcrit en un rapport fonction de l'autre reste ouverte pour ce problème alors que nous y répondons par l'affirmative pour CLIQUE (le problème de la clique maximum). De même, certains résultats d'équivalence entre problèmes ne sont plus valides dans le cadre des rapports dépendant de l'instance, là encore les exemples de STABLE et de CLIQUE sont significatifs. Le formalisme que nous proposons intègre le point de vue des paramètres à l'ensemble des concepts ; il permet ainsi d'envisager ces différentes questions de manière systématique, ne serait-ce déjà que par la possibilité de les poser. Par ailleurs, des travaux mettant en œuvre des analyses de plus en plus fines ne peuvent être directement exposés dans le cadre usuel. Il en résulte une difficulté d'exprimer les résultats, voire même de les identifier et les répertorier de manière précise. Là encore, comme nous le verrons, STABLE fournit un exemple où le manque de formalisme a bloqué les possibilités de rendre compte de résultats dans leur globalité et de les améliorer. Les travaux sur le rapport différentiel ont été directement motivés par la question de la pertinence de la mesure usuelle. Un autre exemple que nous mentionnons dans cet article est celui de la définition d'un *rapport asymptotique*. Ce terme recouvre des analyses qui ne sont valables que lorsque « l'instance tend vers l'infini », en un sens à préciser. Nous montrerons dans la section 4.2 pourquoi la définition usuelle ne nous semble pas pas toujours satisfaisante et en proposerons une autre.

3 Approximabilité – une première classification de NPO

La notion de résolution est implicite dans la définition même d'un problème. On ne peut en effet définir ce dernier ou, plus généralement, une question que par rapport au concept de réponse qu'on adopte. D'un point de vue historique d'ailleurs, le moteur (mais aussi le frein) des travaux sur la décidabilité, la complexité ou plus généralement de toute démarche visant à étudier les limites de l'algorithmique a été la modélisation de la notion d'algorithme⁵ représentant une forme de résolution sur laquelle repose alors la définition des classes de complexité. De même, dans le cadre de l'approximation polynomiale, la notion de problème d'optimisation ne s'impose qu'au terme d'une réflexion sur les buts de cette théorie, notamment à propos de la signification à donner à la *résolution approchée*. Dans l'article [21], nous avons insisté sur le choix du « tout dans la formulation » qui consiste à intégrer dans le problème même les caractéristiques de sa résolution. Nous avons alors montré à quel point, non seulement la fonction objective et les contraintes, mais aussi les valeurs extrémales (optimale et pire) de l'instance sont indispensables pour clarifier le concept de *bonne résolution* en vue d'une évaluation pertinente des algorithmes.

Le choix d'un rapport d'approximation conditionne la manière dont la qualité de résolution d'une instance est prise en compte. Pour étudier la qualité d'un algorithme pour un problème donné, il faut définir un moyen d'agréger les informations pour l'ensemble des instances. Pour cela, la traditionnelle notion d'algorithme approché garantissant une proportion constante de la valeur optimale ne suffit plus comme l'illustrent les exemples détaillés dans ce document. L'intérêt des résultats obtenus avec chaque rapport et les possibilités d'analyses qu'ils offrent nous convainquent de concevoir un formalisme unique permettant de travailler avec l'un ou l'autre des rapports d'approximation. De même, nous avons déjà évoqué le besoin de prendre en compte de manière systématique des résultats d'approximation fonction de l'instance. Cette idée n'est pas neuve en elle-même mais n'a été que très partiellement intégrée au formalisme existant avec les conséquences que nous avons déjà signalées, en particulier sur la mise en retrait

5. Notamment par l'outil des machines de TURING.

de la question des paramètres de l'instance et de leur rôle dans les résultats d'approximation. Une troisième nouveauté concerne des généralisations des algorithmes d'approximation : dans la lignée des schémas d'approximation qui jusque là se singularisaient dans l'outillage de l'approximation polynomiale, nous introduisons l'idée d'étudier le comportement de suites d'algorithmes (appelées *chaînes d'approximation*) qui permettent alors de rendre compte de résultats de convergence. Un bref historique de l'approximation de STABLE (paragraphe 3.1) permet d'illustrer les besoins de recourir à ce type d'outils. Dans la suite de cette section, nous discutons les notions de chaîne d'approximation et de niveau d'approximation. Dans la partie 4 nous creusons alors l'idée de résultats limites en approximation. Nous discutons en particulier les notions asymptotiques présentées comme une forme de « limite par rapport aux instances ».

3.1 Un exemple : une brève histoire de l'approximation du problème de stable maximum

Un bref survol des résultats d'approximation pour STABLE met en évidence l'intérêt de recourir à de nouvelles notions et de poser de nouvelles problématiques. Certains seront repris ou précisés ultérieurement afin d'illustrer un concept ou une problématique. Rappelons en effet que le but de ce travail est de mener une discussion sur les concepts, les résultats ne servant que d'exemples.

Une première famille de résultats concerne des rapports d'approximation fonction du nombre de sommets n de l'instance. Il est évident que, pour le problème non-pondéré, le rapport d'approximation k/n peut être garanti en temps polynomial pour toute constante k : il suffit de chercher de manière exhaustive un meilleur stable parmi les ensembles d'au plus k sommets. Le principal résultat fonction de n pour le cas non-pondéré met en jeu ([10]) un rapport d'approximation de la forme $O(\log^2 n/n)$. Pour le cas pondéré, nous montrons dans [23] comment déduire de ce résultat un algorithme garantissant le rapport $O(\log^2 n/(n \log \log n))$ (proposition 11).

Parallèlement à ces résultats, de nombreux travaux ont porté sur des analyses fonction du degré maximum Δ ou du degré moyen μ ($\mu = 2|E|/n$, où E désigne l'ensembles d'arêtes du graphe). La comparaison de ces deux types de résultats motive une réflexion de fond sur les paramètres en fonction desquels exprimer les résultats d'approximation. Pour étudier cette question, une première condition est de pouvoir l'exprimer dans le formalisme ; nous y reviendrons, notamment dans la partie 6 et au paragraphe 3.3.

Les premiers résultats d'approximation pour STABLE fonction du degré proposent le rapport $1/\Delta$ ([28]) garanti par un algorithme glouton. Nous reprenons, dans le paragraphe 4.2, une analyse de cet algorithme. Cependant, un résultat de TURÁN ([59]) bien antérieur fournit le rapport $1/(\mu + 1)$ pour ce même algorithme, ouvrant ainsi la voie des rapports fonction du degré moyen. Bien que ces résultats fassent désormais partie du lieu commun, ils ont longtemps été les seuls connus et restent relativement compétitifs. La première amélioration significative a été le rapport $2/\Delta$ ([37]) valide également pour le cas pondéré WSTABLE. À partir de 1994, l'approximation de STABLE en termes de degré a connu une évolution rapide. Une analyse fine de l'algorithme glouton a permis d'obtenir ([33]), pour le cas non-pondéré, les rapports $3/(\Delta + 2)$ et $2/(\mu + 2)$ (voir également la proposition 4 du paragraphe 4.2) ; ce dernier rapport peut facilement être amélioré en $5/(2\mu + 3)$ grâce à la très élégante méthode (voir proposition 25) due à NEMHAUSER et TROTTER ([51]) permettant de déterminer une solution semi-intégrale du programme linéaire associé à STABLE. Pour atteindre une approximation similaire pour le cas pondéré, il a fallu développer une toute autre démarche ; le rapport $3/(\Delta + 2)$ est établi dans [31] par une technique de partitionnement du graphe. Dès lors, la version pondérée s'est fortement démarquée, relançant la question des liens entre les versions pondérée et non-pondérée (voir section 6.5) : pour WSTABLE le rapport $3/(\Delta + 2)$ est longtemps resté le meilleur au moment où

le cas non-pondéré bénéficiait de plusieurs améliorations. Mentionnons en particulier le rapport $5/(\Delta + 3) - \epsilon$ ([7]) et même le rapport $O(\log \log \Delta/\Delta)$ ([34])⁶. Toutefois, ces deux approximations ne sont polynomiales que pour des instances de faible degré : les algorithmes correspondant deviennent exponentiels pour des graphes quelconques. Pour le cas général, l'article [34] a été une étape cruciale : son principal résultat est un algorithme de complexité $O(n\Delta^{k-1})$ garantissant, pour une constante k , un rapport de la forme $(6/\Delta) - \epsilon(k) - \eta(\Delta)$ avec $\epsilon(k) \rightarrow 0$ quand $k \rightarrow \infty$ et $\eta(\Delta) \rightarrow 0$ quand $\Delta \rightarrow \infty$. Cependant, le résultat est exprimé de manière très confuse et même imprécise alors que nous verrons qu'il a, dans le formalisme que nous présentons, une forme très simple (proposition 9). Il s'agit d'un exemple significatif où l'inadéquation du cadre classique pour exprimer ces résultats a contraint les auteurs à n'en présenter qu'une forme partielle. Ces différentes approximations correspondent à des améliorations d'un facteur multiplicatif constant qui sont obtenues par des analyses de plus en plus fines. Elles posent implicitement la question de la « meilleure constante » devant le terme $1/\Delta$ ou $1/\mu$. Nous y avons répondu dans [22] en proposant, pour toute constante k , un algorithme de complexité $O(n^{k/2})$ garantissant un rapport de la forme $k/\Delta - o(1/\Delta)$ ainsi qu'un algorithme de complexité $O(n^k)$ garantissant le rapport $\min\{k/\mu, k' \log \log \Delta/\Delta\} - o(1/\mu)$ où k' est une valeur dépendant de k . Là encore, nous verrons comment exprimer ces résultats plus simplement (proposition 9). La question naturelle émanant de ces travaux concerne la possibilité de dépasser l'ordre de grandeur $O(1/\Delta)$. Elle reste à l'étude même si une réponse positive a déjà été établie (théorème 2). En faisant abstraction des détails dans l'expression de ces derniers résultats, leur forme générale se distingue par le rôle que joue k . Il apparaît comme un paramètre de l'algorithme en fonction duquel on peut moduler la qualité de l'approximation avec une incidence directe sur la complexité. Il ne s'agit plus de l'analyse d'un algorithme mais plutôt d'une suite d'algorithmes à l'image d'un schéma d'approximation polynomial. L'idée des chaînes d'approximation est de formaliser ce type de situation, même pour des cas où les rapports d'approximation correspondant ne sont pas du type $1 - \epsilon(k)$ avec $\epsilon \rightarrow 0$ quand $k \rightarrow +\infty$.

3.2 Algorithmes et chaînes d'approximation

Étant donné un problème d'optimisation **NP**-complet, nous avons déjà précisé dans la première section la notion d'*algorithme approché* qui, pour toute instance I de taille n , construit en temps polynomial par rapport à n une solution réalisable. La valeur objective $\lambda(I)$ de cette solution ainsi que les valeurs optimale ($\beta(I)$) et (éventuellement) pire ($\omega(I)$) de I sont essentielles pour l'évaluation (par le biais d'un rapport d'approximation) de la qualité de l'algorithme. Parmi les rapports utilisés, nous avons déjà mentionné $\gamma(I)$ et $\delta(I)$ mais l'étude qui suit est applicable à n'importe quelle mesure d'évaluation. Notre seule restriction est de considérer une mesure $\rho : I \mapsto \rho(I)$ à valeurs entre 0 et 1 d'autant plus proche de 1 que I est bien résolue. Ainsi, un résultat d'approximation consiste à garantir, pour toute instance du problème, un minorant de $\rho(I)$.

La notion de *chaîne algorithmique* généralise celle d'algorithme approché. Le principe est d'étudier globalement le comportement d'une famille d'algorithmes.

Définition 2. Soit Π un problème d'optimisation combinatoire (définition 1) ; on note $\mathcal{I}$ l'ensemble des instances de Π . Soit $f : \mathcal{I} \times \mathbb{N} \rightarrow]0, 1[; (I, k) \mapsto f(I, k)$ une application croissante en k . Une chaîne (algorithmique) d'approximation polynomiale⁷ de rapport f pour Π est une suite d'algorithmes $(A_k)_{k \in \mathbb{N}}$ telle que, pour tout $k \in \mathbb{N}$, A_k est un algorithme approché polynomial pour Π garantissant le rapport d'approximation $\rho_k(I) \geq f(I, k)$, $\forall I \in \mathcal{I}$. ■

6. Ce résultat est l'une des premières référence à un rapport du type $\Omega(1/\Delta)$.

7. Ou simplement *chaîne* lorsqu'aucune confusion n'est possible.

L'intérêt de cette notion est double. D'abord, elle permet de rendre compte globalement d'une famille d'algorithmes et à ce titre permettra de simplifier certains énoncés. Si on se limitait à considérer, pour chaque valeur de k , l'algorithme correspondant et ses garanties, cet intérêt resterait mineur puisqu'un énoncé paramétré par k permet d'exprimer le même résultat de manière aussi simple. Par contre, la simplification devient sensible pour des résultats décrivant le comportement global de la suite d'algorithmes et de la garantie associée qui peut être vue comme une suite de fonctions $k \mapsto f(I, k)$. Cette présentation permet en particulier de définir une notion de convergence par rapport à k rendant compte de garanties correspondant aux grandes valeurs de k . C'est exactement le type de gain que l'on a en utilisant la notion de schéma d'approximation polynomial au lieu d'un algorithme paramétré dont le rapport d'approximation se rapproche de 1 pour certaines valeurs limites du paramètre. Ce type de simplification permet, non seulement de mieux rendre compte des résultats, mais surtout de mieux comparer différentes approximations. Le second intérêt, illustré notamment par l'exemple de STABLE, est de permettre à l'utilisateur de choisir son niveau de qualité et surtout d'expliciter, pour un type de résolution fixé, le coût en complexité d'une amélioration de la qualité.

Dans les prochains paragraphes, nous précisons les notions de niveau d'approximation, à la base de la conception de classes d'approximation, et de convergence.

3.3 Niveaux d'approximation

Nous insistons, une fois de plus, sur le fait qu'un des principaux enjeux de l'approximation polynomiale est de mettre en évidence, au sein de la classe **NPO**, une structure fondée sur les propriétés d'approximation des problèmes. La grande variété de résultats d'approximation suggère de manière très naturelle de hiérarchiser les problèmes en fonction du type de rapport d'approximation qu'on peut garantir. Les classes les plus courantes définies jusqu'à présent sont **APX** et **PTAS**. La première regroupe les problèmes admettant un algorithme à rapport constant et la seconde ceux admettant un schéma d'approximation polynomial; elles vérifient l'inclusion **PTAS** $\subset$ **APX**. On rencontre aussi de manière un peu plus anecdotique les classes **Log-APX** et **Poly-APX** qui recouvrent les problèmes admettant un algorithme à rapport de la forme $g(1/n)$ où n désigne la taille de l'instance et g est respectivement une fonction logarithmique et une fonction polynomiale. Ces deux dernières classes ont été introduites pour pallier au manque d'outils adaptés pour hiérarchiser la région **NPO** $\setminus$ **APX** des problèmes n'admettant pas d'algorithme polynomial à rapport constant. Un premier essai de formalisation est donné dans [44]: la classe **F-APX** comprend les problèmes approximables à un rapport de la forme $g(1/n)$ où n désigne la taille de l'instance et $g \in F$, où F représente une classe de fonctions vérifiant,

$$g \in F \implies \forall (c, h) \in \mathbb{N} \times \mathbb{R}, x \mapsto hg(x^c) \in F \quad (1)$$

De telles restrictions sur la famille de fonctions définissant le niveau sont vérifiées par les classes des fonctions logarithmiques et polynomiales; cependant ces hypothèses excluent certaines classes intéressantes de rapports d'approximation. Mentionnons par exemple le cas de fonctions de la forme $g(x) \in O(x^{1-\epsilon})$, $\epsilon \in]0, 1[$; pour plusieurs problèmes, notamment pour STABLE, le rapport $1/n$ est trivial à garantir alors que sous une hypothèse de complexité, aucun algorithme polynomial ne peut garantir un rapport de la forme $O(1/n^{1-\epsilon})$, $\epsilon \in]0, 1[$ ([36]). Ce type de résultat justifie qu'on ait besoin, entre autres, de différencier des sous-classes de **Poly-APX** en fonction du degré des polynômes, ce qui n'est pas compatible avec l'expression (1). Nous suggérons d'assouplir les règles permettant à une classe de fonctions de définir un niveau d'approximation. La seule règle que nous choisissons d'imposer est la stabilité par passage au minimum (si f et g sont dans F , $\min\{f, g\}$ est dans F). Ceci donne la possibilité, pour garantir un certain niveau d'approximation, de partitionner l'ensemble des instances et garantir ce niveau sur chaque partie,

ce qui est une technique à la base de nombreuses démonstrations.

Un autre inconvénient de la définition de **F-APX** est qu'elle exclut la classe **PTAS** qui est pourtant l'une des plus utilisées en approximation polynomiale. Notons enfin que les classes existantes ne sont conçues que pour le rapport d'approximation classique γ ; pour concevoir un cadre unifié, nous devons préserver la possibilité de moduler les résultats en fonction du rapport.

Pour répondre à ces besoins, nous envisageons la notion de *niveau d'approximation* comme un ensemble de résultats d'approximation. On lui associe alors l'ensemble des problèmes pour lesquels un de ces résultats au moins peut être garanti. étant donné un problème Π d'ensemble d'instances $\mathcal{I}$, on note $\mathcal{F}_\Pi$ (ou seulement $\mathcal{F}$ lorsqu'aucune confusion n'est possible) l'ensemble des rapports possibles pour les chaînes d'approximation polynomiale pour Π , *i.e.*,

$$\mathcal{F}_\Pi = \{f : \mathcal{I} \times \mathbb{N} \rightarrow]0, 1[: (I, k) \mapsto f(I, k)\}.$$

Nous utilisons également la notation $\mathcal{F}'_\Pi$ pour désigner les rapports indépendants de k , *i.e.*,

$$\mathcal{F}'_\Pi = \{f : \mathcal{I} \rightarrow]0, 1[: I \mapsto f(I)\}.$$

Étant donné un rapport d'approximation, à chaque famille de rapports $F \subset \mathcal{F}_\Pi$ vérifiant $\forall (f, g) \in F \times F, \min\{f, g\} \in F$ correspond un *niveau d'approximation* ; toutes les chaînes d'approximation polynomiale garantissant un rapport dans F appartiennent à ce niveau.

Nous évoquons, dans le prochain paragraphe, deux notions de limites de chaînes d'approximation (l'une par rapport au paramètre k et l'autre par rapport aux instances) qui permettent d'affiner la définition de niveau d'approximation.

4 Notions de limites

Certaines analyses ont recours à des notions de limite pour exprimer de manière plus simple un résultat assez technique. L'esprit est toujours le même : il s'agit d'extraire du résultat complet une partie significative plus maniable. Dans le cadre que nous venons d'ébaucher, deux types de limites apparaissent naturellement : l'une se rapportant à une suite d'algorithmes (nous parlons de convergence), et l'autre attachée à l'idée d'« instances limites » (notion asymptotique). La première est directement liée à la notion de chaîne d'approximation ; la seconde a été maintes fois utilisée mais sous des définitions qui méritent, selon nous, commentaires et débats.

4.1 Convergence d'une chaîne d'approximation polynomiale

Dans la littérature, la convergence par rapport à une suite d'algorithmes n'est formalisée que dans le cas de schémas d'approximation mais elle est implicite dans plusieurs résultats récents tels que ceux que nous avons évoqués pour STABLE.

Définition 3. Une chaîne d'approximation polynomiale $(A_k)_{k \in \mathbb{N}}$ converge vers un rapport $\tilde{\rho} \in \mathcal{F}'_\Pi$ si : $\forall \epsilon > 0, \exists \kappa$ tel que $\forall k \geq \kappa$ et $\forall I \in \mathcal{I}, \rho(I, k) = \rho_{A_k}(I) \geq \tilde{\rho}(I)(1 - \epsilon)$. ■

En d'autres termes, $(A_k)_{k \in \mathbb{N}}$ garantit une suite de rapports qui converge uniformément vers $\tilde{\rho}$. Le cas du schéma d'approximation correspond à une chaîne convergeant vers le rapport constant 1 associé à la résolution exacte.

4.2 Notions asymptotiques

Le terme *asymptotique* représente l'idée d'*instances limites* qui mérite d'être discutée et approfondie. De nombreux résultats utilisent ce qualificatif sans référence à une définition précise ; d'ailleurs, la seule définition explicite de rapport asymptotique exclut la plupart de ces résultats. Cependant, le principe consiste toujours à restreindre l'analyse à un groupe d'instances pour

lesquelles un paramètre de référence (la valeur optimale, la taille, le degré (pour un graphe), ...) dépasse un certain seuil. La philosophie sous-jacente est d'associer la valeur de ce paramètre à une mesure de difficulté de l'instance et ainsi, d'une certaine façon, de ne tenir compte que des instances les plus difficiles. Le paramètre joue donc un rôle primordial ; or, il n'est souvent dicté que par des besoins techniques sans justification particulière. Nous proposons, dans cette partie, une analyse critique de la définition usuelle et des résultats qualifiés d'asymptotiques dans la littérature. Cette discussion nous conduit à une nouvelle définition ; d'un côté elle limite le champ de la définition usuelle pour des questions de pertinence et de l'autre offre de nombreuses possibilités nouvelles, permet d'unifier différents résultats de la littérature et les intègre à un cadre précis.

4.2.1 Définition usuelle

Jusqu'ici, la seule notion asymptotique explicitement définie pour un algorithme A est la suivante ([28]) : $\gamma_A^\infty = \sup_{\kappa \geq 0} \inf \{ \gamma_A(I), I \text{ telle que } \beta(I) \geq \kappa \}$. Nous ferons référence à cette définition en parlant de *notion asymptotique par rapport à la valeur optimale*. Elle a été adaptée au cas d'un schéma d'approximation polynomial ([50]) : un schéma d'approximation polynomial asymptotique est une famille A_ϵ d'algorithmes tels que pour chaque $\epsilon \in]0, 1[$, A_ϵ garantit le rapport asymptotique $1 - \epsilon$, i.e., $\forall \epsilon \in]0, 1[, \gamma_{A_\epsilon}^\infty \geq 1 - \epsilon$.

De manière informelle, le rapport asymptotique (meilleur que le rapport constant garanti) décrit le comportement de l'algorithme sur les instances I dont la valeur optimale $\beta(I)$ est élevée. Il revient donc, pour toute constante K , à retirer de l'analyse le comportement des instances de valeur optimale inférieure à K . Si on cherche à justifier intuitivement cette définition, ne s'intéresser qu'aux instances de grande valeur optimale revient à considérer que ce sont les instances les plus « intéressantes ».

4.2.2 Commentaires

La définition de [28] n'est adaptée qu'au cas des rapports constants ; toutefois la généralisation au cas général n'est pas difficile. On dit que le rapport asymptotique ρ est garanti par A (par rapport à la valeur optimale), si $\forall \epsilon, \exists \kappa$ tel que $\forall I$ tel que $\beta(I) \geq \kappa$, $\gamma_A(I) \geq \rho(I)(1 - \epsilon)$. Ce type de définition s'adapte immédiatement au cas d'autres rapports d'approximation.

Hormis ces ajustements, cette définition ne nous semble pas satisfaisante pour deux raisons. D'une part elle exclut de nombreuses situations analogues pour lesquelles $\beta(I)$ est remplacé par un autre paramètre tel que la taille, le degré, ... De tels résultats, relativement courants, sont d'ailleurs souvent qualifiés d'asymptotiques dans la littérature, c'est notamment le cas de certains résultats du paragraphe 3.1 qui distinguent les instances pour lesquelles $\Delta \rightarrow \infty$. Mais d'un autre côté, le critère de la valeur optimale utilisé dans la définition usuelle n'a pas de justification particulière et s'avère parfois discutable. Nous illustrons ces deux inconvénients sur la base d'exemples de résultats et d'analyses concernant BINPACKING et STABLE. Ils nous conduiront à poser la question des paramètres en fonction desquels définir un rapport asymptotique. Au delà de l'exactitude mathématique d'un résultat, il nous semble en effet important de poser la question de sa pertinence. Comme l'illustrent ces exemples, il ne faut pas minimiser l'effet psychologique de la formulation car elle peut impliquer un mauvais usage du résultat ou contribuer à sa mauvaise compréhension.

4.2.2.1 Exemple du *bin-packing*

Rappelons le résultat de FERNANDEZ DE LA VEGA et LUEKER ([27]) : pour tout $k \in \mathbb{N}$, il existe un algorithme polynomial A_k pour BINPACKING garantissant, pour toute instance I , le rapport $\gamma_{A_k}(I) \geq 1 - 1/\beta(I) - 1/k$. Bien qu'il ait été amélioré depuis, ce résultat reste l'un des

plus connus pour ce problème. Un tel rapport d'approximation entre dans le cadre de la définition usuelle d'un schéma d'approximation asymptotique et c'est d'ailleurs ainsi qu'il est le plus souvent référencé. Pourtant, malgré cette terminologie cet algorithme ne se rapproche en rien d'un schéma d'approximation polynomial. Pour nous en convaincre, rappelons un résultat mentionné sans preuve dans [28]. Il fournit un seuil de difficulté qui montre qu'un *schéma d'approximation ne peut être garanti* et que par conséquent le rapport garanti par [27] n'est pas toujours proche de 1. Par ailleurs, la preuve que nous proposons met en évidence que les instances les plus difficiles à approximer sont celles qui ont une petite valeur optimale.

Proposition 1. ([28]) *Si $P \neq NP$, aucun algorithme polynomial ne peut garantir un rapport strictement meilleur que $2/3$ pour BINPACKING.*

Preuve. La preuve consiste en une réduction du problème de PARTITION ([28], page 61) qui est NP-complet. Pour une liste de $x_1, \dots, x_n$ de n rationnels, il consiste à décider si on peut la partager en deux listes disjointes $\{x_i, i \in I\}$ et $\{x_j, j \in J\}$ (avec $\{1, \dots, n\} = I \cup J$) telles que les sommes des nombres de chaque liste coïncident : $\sum_{i \in I} x_i = \sum_{j \in J} x_j$.

En divisant chaque nombre par la demi-somme totale $\tilde{x}_i := x_i(1/2) \sum_{j=1}^n x_j$, le problème revient à décider si les nombres $\tilde{x}_i, i \in \{1, \dots, n\}$ peuvent être rangés en deux boîtes de capacité 1. Il s'agit donc de décider si la liste $\tilde{L} = (\tilde{x}_i, i \in \{1, \dots, n\})$, vue comme instance de BINPACKING, a pour valeur optimale 2 ou strictement plus. Mais alors, un algorithme d'approximation pour BINPACKING garantissant un rapport $\rho > 2/3$ permettrait de répondre à cette question : si la valeur optimale de $\tilde{L}$ est égale à 2, un tel algorithme retournerait une solution de valeur strictement inférieure à 3, donc de valeur 2. Par contre, dans le cas inverse toute solution réalisable, donc en particulier celle retournée par l'algorithme, serait de valeur au moins égale à 3. ■

Remarquons que pour toute instance I de BINPACKING de valeur optimale $\beta(I) \geq 4$, le résultat de [27] permet d'obtenir un rapport d'approximation strictement meilleur que $2/3$. Les instances bloquant les possibilités d'approximation sont donc celles de valeur optimale inférieure à 3. Énoncer le résultat de [27] comme un schéma d'approximation asymptotique nous paraît alors réducteur puisque cette formulation ne rend pas compte des instances difficiles (de faible valeur optimale).

Pour renforcer encore notre propos, considérons la classe $\mathcal{C}$ d'instances de BINPACKING correspondant aux listes croissantes de rationnels de la forme $(x_1, \dots, x_k, a, b, 1, \dots, 1)$ avec $\sum_{i=1}^k x_i \leq 1$ et $a + b > 1$; $\mathcal{C}$ est bien sûr reconnaissable en temps polynomial. Appelons $\text{BINPACKING}_{\mathcal{C}}$ le problème de BINPACKING restreint aux listes de $\mathcal{C}$. On a alors le résultat suivant.

Proposition 2. $\text{BINPACKING}_{\mathcal{C}}$ est NP-difficile.

Preuve. Reprenons la démonstration de la NP-difficulté du problème de PARTITION ([28]) qui résulte d'une réduction polynomiale du problème 3-DM (couplage tri-dimensionnel défini dans la section A). Connaître cette réduction n'est pas nécessaire pour la compréhension de la preuve ; son principe, que nous rappelons ici, suffit pour la suite. À partir d'une instance I_1 de 3-DM comprenant k triplets on construit polynomialement $k+1$ entiers (en ordre croissant) $a_1, \dots, a_k, B$ (avec $B \leq \sum_{i=1}^k a_i$) tels que I_1 est admissible si et seulement si il existe une partie $A' \subset \{1, \dots, k\}$

telle que $B = \sum_{i \in A'} x_i$. Posons alors :

$$\left\{ \begin{array}{lcl} a & = & 2 \sum_{i=1}^k a_i - B \\ b & = & \sum_{i=1}^k a_i + B \\ K & = & 2 \sum_{i=1}^k a_i \\ \tilde{a}_i & = & \frac{a_i}{K} \quad \forall i \in \{1, \dots, k\} \\ \tilde{a} & = & \frac{a}{K} \\ \tilde{b} & = & \frac{b}{K} \end{array} \right.$$

Il est clair qu'étant donnés $A' \subset \{1, \dots, k\}$ et $\bar{A}' = \{1, \dots, k\} \setminus A'$, le fait que $\sum_{i \in A'} x_i = B$ implique

$$\tilde{a} + \sum_{i \in A'} \tilde{x}_i = \tilde{b} + \sum_{i \in \bar{A}'} \tilde{x}_i = 1$$

Par ailleurs, $\tilde{a} + \tilde{b} > 1$. La liste $(\tilde{x}_1, \dots, \tilde{x}_k, \min\{\tilde{a}, \tilde{b}\}, \max\{\tilde{a}, \tilde{b}\})$ est donc une instance de BINPACKING_C . Sa valeur optimale est 2 si et seulement si l'instance initiale de 3-DM est admissible et 3 sinon. Résoudre cette instance de BINPACKING_C permettrait de résoudre l'instance initiale de 3-DM. Ce dernier problème étant **NP**-complet, BINPACKING_C est **NP**-difficile. ■

Par le même argument que dans la proposition 1 on montre que sous l'hypothèse $\mathbf{P} \neq \mathbf{NP}$, BINPACKING_C n'admet pas d'algorithme polynomial d'approximation garantissant un rapport strictement meilleur que $2/3$.

Par contre, étant donnée une instance I de BINPACKING_C , si on note p le nombre de « 1 », la valeur optimale de I est soit $2 + p$, soit $3 + p$ selon que les nombres $x_1, \dots, x_k, a, b$ peuvent tenir dans 2 ou 3 boîtes. Ajouter des « 1 » ne rend l'instance ni plus simple, ni plus difficile puisque chacun constituera une boîte à lui seul dans toute solution réalisable. Ici, la valeur de l'optimum n'est donc nullement liée à la difficulté de l'instance. Par contre, tout algorithme mettant les k premiers nombres dans la même boîte garantit le rapport $(p + 2)/(p + 3)$, i.e., le rapport asymptotique 1. On en déduit le résultat suivant.

Proposition 3. *Si $\mathbf{P} \neq \mathbf{NP}$, BINPACKING_C n'admet pas d'algorithme polynomial d'approximation garantissant un rapport strictement meilleur que $2/3$. Par contre, il admet un algorithme polynomial asymptotiquement optimal (par rapport à la valeur optimale).*

Pour les problèmes BINPACKING et BINPACKING_C , l'aspect asymptotique par rapport à la valeur optimale revient donc à éliminer les instances bloquant les possibilités d'approximation pour obtenir des résultats asymptotiques dépassant radicalement les niveaux de qualité qu'on peut obtenir pour les instances générales.

4.2.2.2 Exemple du problème de stable maximum

Pour d'autres problèmes, au contraire, les instances de grande valeur optimale sont les plus difficiles et les résultats asymptotiques qui s'y rapportent permettent d'exprimer des résultats qui représentent assez fidèlement le comportement pour toutes les instances. Considérons en particulier la classe très large des problèmes consistant à sélectionner de manière optimale une partie dans un ensemble $\mathcal{E}$ de taille n : les problèmes STABLE ou CLIQUE ou encore H-TRANSVERSAL et G-TRANSVERSAL sont de ce type. Pour les problèmes de minimisation de cette forme, les instances dont la valeur optimale est majorée par une constante k sont reconnaissables et résolubles en temps polynomial : il suffit, pour les reconnaître, d'énumérer toutes les parties de taille k de $\mathcal{E}$ pour s'assurer qu'une au moins est réalisable. Pour les résoudre, il suffit d'énumérer toutes les

parties de $\mathcal{E}$ de taille au plus k . Un raisonnement analogue s'applique à certains problèmes de maximisation, en particuliers ceux pour lesquels toute partie d'une solution réalisable est réalisable : il suffit alors, pour vérifier que la valeur optimale est au plus k , de vérifier qu'aucune partie de taille $k + 1$ n'est réalisable. La résolution de telles instances est, quant à elle, identique au cas de la minimisation.

Le problème STABLE est de ce type. Nous proposons ci-dessous une analyse de l'algorithme glouton pour ce problème qui montre à quoi correspond, dans ce cas, un résultat asymptotique (par rapport à la valeur optimale). Dans ce qui suit, pour $v \in V$, nous notons $\Gamma(v)$ l'ensemble de sommets adjacents à v (voisins de v) et par $\delta(v)$ la quantité $|\Gamma(v)|$ (le degré du sommet v).

```

BEGIN *GLOUTON*
  S ← ∅;
  REPEAT
    v ← argminvi ∈ V {δ(vi)};
    S ← S ∪ {v};
    V ← V \ ({v} ∪ Γ(v));
    G ← G[V];
  UNTIL V = ∅;
END. *GLOUTON*
```

De nombreuses analyses existent pour mettre en évidence des garanties de performances pour cet algorithme (voir notamment [33]). Parmi celles-ci, nous en reprenons ici une relativement naïve mais qui illustre bien notre propos. Remarquons d'abord qu'il suffit de faire l'analyse sur une composante connexe : si un rapport ρ s'applique à chaque composante, il est clair qu'il s'applique alors au graphe entier en remarquant que les valeurs approchées et optimales des différentes composantes s'ajoutent pour constituer respectivement la valeur approchée et la valeur optimale du graphe entier. Considérons donc un graphe connexe. De même, en notant S^* un stable optimal, on peut se ramener au cas où $S \cap S^* = \emptyset$ en faisant la remarque suivante :

$$\frac{|S| - |S \cap S^*|}{|S^*| - |S \cap S^*|} \leq \frac{|S|}{|S^*|}.$$

Afin de majorer le nombre d'arêtes adjacentes au stable S remarquons qu'au plus Δ arêtes sont adjacentes au premier sommet mis dans la solution et au plus $\Delta - 1$ aux suivants (d'après la connexité et le critère glouton). Au plus $|S|(\Delta - 1) + 1$ arêtes sont donc adjacentes à S . Par ailleurs, chaque sommet de S^* est lié à au moins un sommet de S en utilisant, d'une part le fait que $S \cap S^* = \emptyset$ et d'autre part le fait que S est maximal pour l'inclusion. On en déduit :

$$|S|(\Delta - 1) + 1 \geq |S^*| \iff \frac{|S|}{|S^*|} \geq \frac{1}{\Delta - 1} \left(1 - \frac{1}{|S^*|}\right) \quad (2)$$

Cette expression correspond au rapport asymptotique (au sens de la valeur optimale) étendu au cas d'un rapport non constant. Il a alors une interprétation complètement différente que pour BINPACKING. En effet, sachant que les graphes pour lesquels $|S^*| \leq k$ (où k est une constante arbitraire) peuvent être résolus en temps polynomial, l'algorithme peut être amélioré de la manière suivante. On commence par rechercher un stable de taille k ; s'il n'en existe pas on recherche exhaustivement un stable optimal, sinon ($|S^*| \geq k + 1$) on détermine une solution gloutonne. L'analyse précédente fournit alors, pour cet algorithme, la garantie :

$$\frac{1}{\Delta - 1} \left(1 - \frac{1}{k + 1}\right) \quad (3)$$

En d'autres termes, la garantie asymptotique fournie par l'expression (2) peut artificiellement être modifiée (modulo une plus forte complexité) en une chaîne à rapport convergeant vers $1/(\Delta - 1)$

(expression (3)). Dire que l'algorithme garantit le rapport asymptotique $1/(\Delta - 1)$ est alors essentiellement une simplification technique permettant d'exprimer la partie la plus significative du rapport garanti.

4.2.2.3 Autres paramètres

Avant même de prendre parti sur la signification à accorder à un résultat asymptotique, il nous semble gênant d'avoir, sous une même terminologie, des situations aussi différentes. Dans le cas de BINPACKING les résultats asymptotiques consistent à considérer des classes d'instances « plus faciles » pour obtenir des résultats meilleurs que ce qui est possible pour le cas général. Pour le cas de STABLE, ils ne servent qu'à exprimer de manière plus simple un résultat en extrayant « sa partie prépondérante ». Nous suggérons de ne retenir que ce second type de résultats asymptotiques sachant que dans l'autre cas, l'expression complète du rapport d'approximation fournit une information intéressante, notamment sur la manière dont il évolue en fonction de la valeur optimale. En se restreignant aux instances difficiles, un résultat asymptotique exploite alors les simplifications techniques que procurent ces instances sans affecter la teneur du résultat.

Mais pourquoi se restreindre alors aux expressions fondées sur la valeur optimale et ne pas prendre en compte d'autres types de « limites d'instances » ? Dans le cas de BINPACKING, un rapport du type $\rho(1 - 1/n)$ pourrait tout à fait être lu comme un *rapport asymptotique* ρ . On pourrait notamment lui appliquer la remarque faite pour STABLE : les instances comportant un nombre borné d'objets sont résolubles en temps polynomial de sorte qu'un résultat du type $\rho(1 - 1/n)$ pourrait être transformé en une chaîne d'approximation à rapport convergeant vers ρ .

Revenons maintenant à l'analyse de l'algorithme glouton pour STABLE. Par un raisonnement similaire n'exploitant pas le critère glouton on peut établir, pour tout stable $\tilde{S}$ maximal pour l'inclusion la garantie

$$\frac{|\tilde{S}|}{|S^*|} \geq \frac{1}{\Delta} = \frac{1}{\Delta - 1} \left(1 - \frac{1}{\Delta}\right) \quad (4)$$

Le cas des graphes tels que $|S^*| < \Delta$ peut être écarté de telles analyses car un raisonnement très simple permet alors d'établir que l'algorithme glouton garantit le rapport $2/(\Delta - 1)$. Si $|S^*| \geq \Delta$, l'analyse que nous avons faite pour l'algorithme glouton fournit logiquement un meilleur rapport que celle valable pour tout stable maximal. Une différence importante entre ces deux résultats réside dans le fait qu'on ne peut pas résoudre de manière optimale le cas des graphes pour lesquels $\Delta \leq k$. Le problème STABLE pour de tels graphes est **NP**-complet dès que $k \geq 3$ ([28]). Toutefois, considérer ce résultat comme asymptotique « par rapport à Δ » ne serait pas absurde sachant que, si STABLE reste difficile pour les graphes de degré borné, il peut quand même être approché avec un rapport constant pour ces graphes alors qu'une telle garantie est exclue pour le problème général. Bien sûr, pour cet exemple, considérer le rapport asymptotique n'a pas d'intérêt puisque l'expression exacte $(1/\Delta)$ du rapport - qui est toujours préférable - est plus simple que le rapport asymptotique (expression (4)) qui en résulte.

Considérons maintenant l'algorithme de recherche locale 2-OPT pour STABLE. Dans le cadre de la recherche d'un stable maximum, les améliorations élémentaires usuelles sont les *a-améliorations* qui consistent, pour $a \in \mathbb{N}$, à retirer $a' < a$ sommets d'un stable S et ajouter $a' + 1$ sommets de $V \setminus S$ en préservant la structure de stable. Lorsqu'aucune 1-amélioration n'est possible, le stable est maximal. Une 2-amélioration consiste, soit en une 1-amélioration ($a' = 0$), soit en le retrait d'un sommet et l'ajout de deux. Dans ce qui suit, nous supposons qu'un stable initial est calculé par application du GLOUTON.

BEGIN *2-OPT*

$S \leftarrow \text{GLOUTON}(G);$

WHILE $\exists$ 2-amélioration (v_1, v_2, u) de S **DO** $S \leftarrow S \cup \{v_1, v_2\} \setminus \{u\}$ **OD**

OUTPUT S ;
 END. *2-OPT*

La proposition suivante est établie dans [34]. Nous mentionnons néanmoins la preuve qui montre bien la différence avec les analyses précédentes.

Proposition 4. ([34]) *2-OPT* garantit le rapport $2/(\Delta + 1)$. Si $|S^*| \leq n/2$, ce rapport peut être amélioré en $3/(\Delta + 2)$.

Preuve. Appelons S la solution construite et S^* une solution optimale. Comme pour l'analyse de l'algorithme glouton, on peut se ramener au cas où $S^* \cap S = \emptyset$. On décompose alors S^* en $S_1^* \cup S_2^*$ où $S_1^* = \{s \in S^* : |\Gamma(s) \cap S| = 1\}$ (les sommets de S^* n'ayant qu'un voisin dans S). Un sommet de S ne peut avoir deux voisins dans S_1^* , sinon l'algorithme aurait eu une étape de plus. On en déduit $|S| \geq |S_1^*|$. L'ensemble S étant maximal pour l'inclusion, le nombre d'arêtes issues de S est au moins égal à $|S_1^*| + 2|S_2^*| + n - |S^*| - |S|$. On en déduit :

$$\Delta|S| \geq 2|S^*| - |S| + n - |S^*| - |S| \quad (5)$$

Le rapport $2/(\Delta + 1)$ se déduit immédiatement de l'expression (5) en posant $n - |S^*| - |S| \geq 0$. Le rapport $3/(\Delta + 2)$ se déduit de l'expression (5) en posant $n \geq 2|S^*|$ (nous verrons, dans la section 6 comment transférer ce résultat à tout graphe). ■

Dans ce cas, parler de « rapport asymptotique $2/\Delta$ et $3/\Delta$ » a déjà plus de sens et permet de mieux comparer ces deux résultats. Enfin, les approximations que nous avons évoquées dans la partie 3.1 qui mettent en jeu des rapports de la forme $k/\Delta - o(1/\Delta)$ illustrent également l'intérêt de s'intéresser à des notions asymptotiques fondées sur le degré, au moins pour ce problème. Nous montrerons dans la section 5 les énoncés simplifiés de ces résultats.

La définition que nous proposons dans le paragraphe 4.2.4 donne la possibilité de considérer différents types de paramètres et fixe un cadre d'interprétation des résultats asymptotiques permettant de les comparer. Le paramètre intervenant dans la définition doit mesurer, d'une manière ou d'une autre, la difficulté des instances de sorte que considérer les situations limites où ce paramètre tend vers l'infini réponde à l'objet que nous nous sommes fixé.

4.2.3 Ordre de difficulté

La notion d'ordre de difficulté a cette vocation. Elle fait référence à un niveau d'approximation car il est naturel de considérer que la difficulté (au sens de l'approximation) dépend du type d'approximation pris comme référence.

Définition 4. Soient un problème Π , $\mathcal{I}$ son ensemble d'instances et un niveau d'approximation $\mathcal{F}$ pour Π tel que Π n'appartient pas à ce niveau (*i.e.*, il n'existe pas, sous une hypothèse du type $\mathbf{P} \neq \mathbf{NP}$, de chaîne polynomiale pour Π garantissant un rapport dans $\mathcal{F}$). Alors une application $d : \mathcal{I} \rightarrow \mathbb{N}$ est un ordre de difficulté pour Π par rapport au niveau $\mathcal{F}$, si, $\forall M \in \mathbb{N}$, la restriction de Π aux instances $I \in \mathcal{I}, d(I) \leq M$ appartient au niveau $\mathcal{F}$. ■

Les instances de faible degré sont donc mieux approximables que les générales, ce qui permet de voir d comme un indicateur de difficulté pour ce niveau d'approximation.

Nous donnons ci-après des exemples immédiats illustrant cette notion et montrant que plusieurs concepts très classiques de théorie de la complexité peuvent s'exprimer en termes d'ordre de difficulté. Dans le prochain paragraphe, nous l'utilisons pour définir une limite par rapport aux instances. Mais l'intérêt de cet outil dépasse largement la définition d'un cadre asymptotique ; en particulier, il permet de structurer les instances d'un problème en vue de mieux comprendre ce qui conditionne leur difficulté. Nous reviendrons, dans la section 7, sur cet aspect qui fait intervenir essentiellement la notion d'ordre de difficulté ainsi que celle de réduction discutée dans la section 6. Nous développerons alors des exemples plus techniques qui risqueraient, dans ce paragraphe, de nuire à la fluidité du document.

Plusieurs notions traditionnelles de complexité sont liées à la définition 4. Pour tout problème d'optimisation **NP**-difficile, la taille de l'instance constitue un ordre de difficulté pour le niveau de la résolution exacte (*i.e.*, pour le niveau d'approximation correspondant au rapport constant égal à 1). De même, les problèmes pseudo-polynomiaux sont définis dans [28] à partir d'une fonction $\max : I \mapsto \max\{I\}$ représentant la valeur du plus grand nombre intervenant dans I ; par exemple, dans le cas de problèmes pondérés, $\max$ est la valeur du plus grand poids. Les problèmes **NP**-complets pseudo-polynomiaux s'interprètent comme les problèmes pour lesquels $\log \max$ est un ordre de difficulté pour la résolution exacte. Pour les problèmes STABLE et COLORATION, le degré maximum est un ordre de difficulté par rapport au niveau d'approximation des rapports (classiques) constants. Il en est de même pour la taille du plus grand ensemble dans H-TRANSVERSAL.

Avant d'aller plus loin dans l'analyse de cette notion, nous nous intéressons, dans le prochain paragraphe, à son utilisation pour définir un cadre cohérent de notions asymptotiques.

4.2.4 Proposition de définition

Le concept d'ordre de difficulté nous permet de définir une notion de limite des instances. Remarquons que dans la définition 4, le fait que Π n'appartient pas⁸ au niveau $\mathcal{F}$ permet d'établir le résultat suivant.

Proposition 5. *Soit un problème Π d'ensemble d'instances $\mathcal{I}$ et d un ordre de difficulté pour Π par rapport à un niveau d'approximation $\mathcal{F}$. On a : $\sup_{I \in \mathcal{I}} d(I) = +\infty$.*

Nous proposons de définir des notions asymptotiques en considérant les instances ayant un grand degré de difficulté. Étant donné un problème Π et un degré de difficulté d , un résultat d'approximation asymptotique se rapporte aux instances I vérifiant $d(I) \geq D$, où D est une constante arbitrairement grande. Cette notion dépend du degré de difficulté choisi mais permet une définition générique pour l'ensemble des problèmes.

La démarche consiste donc à exclure les instances dont le degré de difficulté ne dépasse pas D ; par définition, de telles instances sont plus faciles à approximer que le problème général, ce qui justifie la pertinence de ce point de vue dans le cadre de l'approximation. Intuitivement, le résultat, qualifié d'asymptotique, peut s'interpréter comme la « partie prépondérante » d'un résultat plus lourd à exprimer pour l'ensemble des instances. Sous ce point de vue, la partie « négligée », ou plutôt non spécifiée, correspond à des instances mieux approximables, ce qui justifie qu'on s'autorise, dans la formulation du résultat, à se focaliser sur les instances de grand degré de difficulté. La proposition 6 formalise cette intuition.

Définition 5. étant donné un ordre de difficulté (associé à un niveau d'approximation $\mathcal{F}$) $d : \mathcal{I} \rightarrow \mathbb{N}$ des instances d'un problème Π , on dit qu'une chaîne algorithmique $(A_k)_{k \in \mathbb{N}}$ pour Π a un *rapport asymptotique* $f \in \mathcal{F}_\Pi$ si : $\forall \epsilon > 0$ et $\forall k \in \mathbb{N}$, $\exists D$ tel que $\forall I \in \mathcal{I}$ avec $d(I) \geq D$, $\rho_k(I) \geq f(I, k)(1 - \epsilon)$. ■

Dans le cadre plus restreint des algorithmes approchés, la définition équivalente devient ($f \in \mathcal{F}'_\Pi$) : $\forall \epsilon > 0$, $\exists D$ tel que $\forall I \in \mathcal{I}$ avec $d(I) \geq D$, $\rho(I) \geq f(I)(1 - \epsilon)$, si bien que la définition 5 correspond à une suite de rapports asymptotiques.

Les notions de rapport asymptotique d'un algorithme et de schéma d'approximation asymptotique définies dans [28, 50] correspondent au cas où le degré de difficulté est la valeur optimale. L'intérêt de notre définition est d'abord d'offrir un cadre permettant l'usage d'autres degrés de difficulté. Nous verrons dans les prochains paragraphes qu'elle permet de rendre compte de nombreux résultats qualifiés jusque là d'asymptotiques sans définition précise de ce terme. Mais elle a aussi le mérite de mettre l'accent sur le débat concernant la pertinence des résultats asymptotiques. Conformément au choix que nous avons fait, cette définition exclut du cadre asymptotique

8. Sous une hypothèse du type $\mathbf{P} \neq \mathbf{NP}$.

certain résultats qualifiés jusqu'ici comme tels mais qu'il ne nous semble pas opportun d'évoquer en ces termes afin d'en éviter de mauvaises interprétations. C'est le cas de BINPACKING et du résultat à rapport $1 - 1/\beta(I) - 1/k$ que nous avons déjà discuté. Du point de vue de notre définition, on ne peut plus parler de *schéma d'approximation asymptotique*. Nous avons délibérément choisi de limiter l'usage de résultats asymptotiques à la simplification d'énoncés complexes par la mise en évidence de leur partie prépondérante. La proposition suivante découle immédiatement de la définition et généralise une remarque que nous avons faite au paragraphe 4.2.2. Elle illustre qu'un résultat asymptotique fournit bien une représentation, simplifiée mais fidèle, de résultats plus lourds.

Proposition 6. *En adoptant les notations de la définition 5 :*

1° si Π admet une chaîne d'approximation avec un rapport asymptotique $f \in \mathcal{F}'$ par rapport à un ordre de difficulté associé au niveau $\mathcal{F}$, alors pour tout $\epsilon > 0$, il existe une chaîne à rapport dans $\mathcal{F} \cup (1 - \epsilon)\mathcal{F}'$, où $(1 - \epsilon)\mathcal{F}' = \{(1 - \epsilon)f' : f' \in \mathcal{F}'\}$;

2° si Π admet un algorithme d'approximation avec un rapport asymptotique f , et si $\forall g \in \mathcal{F}$, $g \geq f$, alors Π admet une chaîne d'approximation convergeant vers f .

Le principal cas où la condition $\forall g \in \mathcal{F}$, $g \geq f$ est satisfaite est $\mathcal{F} = \{1\}$ (correspondant à la résolution exacte). Cependant, dans l'expression $\mathcal{F} \cup (1 - \epsilon)\mathcal{F}'$, $\mathcal{F}'$ constitue la composante prépondérante sachant que $\mathcal{F}$ est, dans ce type de résultat, un niveau d'approximation « meilleur » que $\mathcal{F}'$.

4.3 Convergence asymptotique

Mentionnons enfin la possibilité de mixer les deux notions de limite. Parmi les différentes manières de le faire, la *convergence asymptotique*, présentée dans la définition suivante, est la seule notion mixte qui nous a jusqu'ici été utile.

Définition 6. Étant donné un ordre de difficulté (associé à un niveau d'approximation $\mathcal{F}$) $d : \mathcal{I} \rightarrow \mathbb{N}$ des instances d'un problème Π , on dit qu'une chaîne d'approximation polynomiale $(A_k)_{k \in \mathbb{N}}$ converge asymptotiquement vers un rapport $\tilde{\rho} \in \mathcal{F}'_{\Pi}$ si : $\forall \epsilon > 0$, $\exists \kappa$ tel que $\forall k \geq \kappa$, $\exists D$ tel que $\forall I \in \mathcal{I}$ avec $d(I) \geq D$, $\rho(I, k) = \rho_{A_k}(I) \geq \tilde{\rho}(I)(1 - \epsilon)$. ■

Cette définition est équivalente à dire que $(A_k)_{k \geq 0}$ garantit une suite de rapports asymptotiques qui convergent uniformément vers $\tilde{\rho}$. Dans le cas où $\tilde{\rho}$ est la fonction constante égale à 1 et où $d = \beta$, on retrouve la définition usuelle d'un schéma d'approximation asymptotique (voir paragraphe 4.2.1).

5 Exemples

De nombreuses connexions existent entre les concepts que nous avons présentés dans les deux précédentes sections. C'est pourquoi nous regroupons ici quelques exemples significatifs de leur utilisation qui illustrent leur intérêt. Ainsi, nous ne citons pas ces résultats pour eux-même ni pour illustrer le type d'algorithmes auxquels ils se rapportent. Nous les avons sélectionnés de manière à mettre en évidence leur lien avec le formalisme que nous avons décrit et les différentes questions qu'ils soulèvent. Cet objectif justifie leur classement par type, par référence aux notions auxquelles ils se rapportent. Ce plan nous amène donc à évoquer les résultats relatifs à un problème donné sans chercher à respecter leur chronologie ni un ordre dans leur qualité.

Comme nous l'avons déjà signalé, l'intérêt de la définition de *niveau d'approximation* est qu'elle permet d'associer, de manière unifiée, la plupart des résultats d'approximation à des classes de problèmes.

5.1 Algorithmes d'approximation

Dans le formalisme des chaînes d'approximation, les algorithmes d'approximation correspondent au cas des rapports f indépendants de k :

$$\exists \tilde{f} : \mathcal{I} \rightarrow]0, 1[, \forall I \in \mathcal{I}, \forall k \in \mathbb{N}, f(I, k) = \tilde{f}(I)$$

(f peut alors être assimilée à un élément de $\mathcal{F}'_{\Pi}$) et on peut, sans perte de généralité, restreindre la chaîne d'approximation à un unique algorithme **A** en sélectionnant l'un des A_k .

5.1.1 Classe **APX**

NPO peut être structuré en fonction de la dépendance de $\tilde{f}$ par rapport au paramètre I . Si $\tilde{f}$ est une constante universelle, on retrouve le niveau **APX** des algorithmes à rapport γ constant.

Citons par exemple l'algorithme de couplage maximal pour le problème **G-TRANSVERSAL** qui consiste à prendre tous les sommets couverts par un couplage maximal. Il s'agit bien d'une couverture de sommets : le couplage étant maximal, toute arête du graphe est adjacente à au moins une de ses arêtes. Par ailleurs, une couverture doit nécessairement contenir au moins une extrémité de chaque arête du couplage (pour couvrir ces arêtes). Cet algorithme polynomial garantit donc le rapport $\gamma = 1/2$, c'est à dire **G-TRANSVERSAL** $\in$ **APX**.

Le niveau **APX** peut lui-même être divisé en sous-niveaux en fonction des valeurs de la constante.

5.1.2 Rapports dépendant de l'instance

De nombreux problèmes sont connus comme ne faisant pas partie de **APX**⁹. En particulier, si **P** $\neq$ **NP**, ni **STABLE**, ni **CLIQUE** ([3]), ni **COLORATION** ([47]) ne sont dans **APX**. De tels résultats justifient qu'on s'intéresse à des rapports $\tilde{f}$ non minorés (pour lesquels $\inf_{I \in \mathcal{I}} \tilde{f}(I) = 0$), on retrouve la notion de rapport d'approximation dépendant de l'instance. Le principal intérêt d'avoir une gamme riche de niveaux d'approximation est d'ailleurs de pouvoir distinguer différentes couches d'approximabilité dans **NPO** $\setminus$ **APX**.

Les deux principales couches usuelles sont **Log-APX** et **Poly-APX** qui correspondent respectivement, au cas où $\tilde{f}$ est une fonction logarithmique et polynomiale par rapport à la taille du problème. L'un des premiers problèmes à avoir été identifié dans **Log-APX** est **H-TRANSVERSAL** (cf., [40] pour sa version non-pondérée et [11] pour sa version pondérée). Il montre l'intérêt de pouvoir subdiviser ce niveau d'approximation puisque le rapport $1/(1 + \log n)$ peut être garanti alors qu'il existe une constante $0 < c < 1$ telle que si **P** $\neq$ **NP**, aucun algorithme polynomial ne peut garantir le rapport $1/c \log n$ ([56]).

La classe **Poly-APX** contient de très nombreux problèmes et en particulier **COLORATION**, **STABLE**, **CLIQUE**. Là encore, comme nous l'avons déjà signalé dans le paragraphe 3.3, on a besoin de différencier cette classe en fonction du degré du polynôme intervenant dans l'expression du rapport d'approximation. Les classes usuelles ne permettent pas cette distinction ; pourtant les trois problèmes que nous venons de citer ne peuvent, sous l'hypothèse **NP** $\neq$ **ZPP**¹⁰, être approchés avec un rapport du type $O(1/n^{1-\epsilon})$ ([36, 26]) mais admettent par contre trivialement le rapport $1/n$. Ceci justifie qu'on s'intéresse, pour ces problèmes, à des rapports du type $f(n)/n$ où $\forall \epsilon > 0, f(n) = o(n^\epsilon)$.

9. Sous une hypothèse de complexité.

10. **ZPP** désigne la classe des problèmes résolus sans erreur par un algorithme non-déterministe de complexité moyenne polynomiale ; **P** $\subset$ **ZPP** $\subset$ **NP**.

Les résultats actuels pour ces trois problèmes correspondent à des cas où f est une fonction poly-logarithmique de sorte de tels niveaux d'approximation semblent de plus en plus intéressants à considérer.

Théorème 1.

- STABLE et CLIQUE sont approximables à rapport $O(\log^2 n/n)$ ([10]).
- COLORATION est approximable à rapport $O(\log^3 n/(n \log^2 \log n))$ ([30]).

5.1.3 Rapports fonction d'autres paramètres

Une question importante que notre travail permet de considérer de manière systématique est la *problématique des paramètres*¹¹. Elle consiste à envisager différents types de résultats non-constants en fonction des paramètres de l'instance qui interviennent dans l'expression du rapport. En effet, si les outils classiques n'envisagent que des rapports fonction de la taille du problème, de nombreux résultats d'approximation (voir par exemple le paragraphe 4.2.2) font intervenir d'autres paramètres de l'instance, notamment le degré maximum pour des problèmes de graphes. Enrichir le formalisme n'est pas absolument indispensable pour établir de tels résultats; par contre une telle extension a d'abord l'intérêt de mettre en évidence l'existence de cette question et surtout offre un cadre adéquat pour envisager des études autour de cette problématique. Dans ce cadre, il nous semble en particulier intéressant d'étudier dans quelle mesure, pour un problème donné, le choix du paramètre conditionne le résultat. C'est notamment une manière d'étudier quel paramètre conditionne la difficulté d'une instance. Le fait que le formalisme classique n'intègre pas la possibilité d'exprimer les rapports d'approximation en fonction de différents paramètres est autant le signe que cette question n'a pas été envisagée de manière systématique que le frein à son étude.

Pour illustrer le type de résultats auquel cette question peut conduire, mentionnons le cas de CLIQUE.

Proposition 7.

- 1° Si un rapport d'approximation $\gamma(n)$ peut être garanti en temps polynomial pour CLIQUE, alors le rapport $\gamma(\Delta + 1)$ peut également être garanti (et inversement).
- 2° CLIQUE est approximable au niveau $O(\log^2 \Delta/\Delta)$.

Le point 2 se déduit immédiatement du point 1 et du théorème 1. Le point 1, quant à lui, sera montré dans la prochaine section pour illustrer le concept de *réduction*. Il exprime le fait que les paramètres n et Δ sont équivalents pour ce problème. Nous montrons, dans la partie 6, que ce résultat n'est pas vrai pour STABLE. Ceci illustre que CLIQUE est plus simple que STABLE. Une autre facette de ce phénomène est que le problème CLIQUE, contrairement à STABLE, devient polynomial pour les graphes à degré borné. En d'autres termes, Δ est un ordre de difficulté pour CLIQUE par rapport à la résolution exacte.

D'ailleurs, souvenons-nous (cf., 3.1) que deux familles de résultats d'approximation ont été développées pour STABLE, l'une fonction du paramètre n et l'autre fonction des paramètres Δ (degré maximum) et μ (degré moyen). Elles sont radicalement différentes de deux points de vue. Elles diffèrent d'abord dans la forme des rapports d'approximation : au résultat du théorème 1 a longtemps été opposée une suite de résultats du type k/Δ ou k/μ pour des valeurs particulières de k . Mais le plus surprenant est que ces deux familles de résultats diffèrent surtout par l'étalement des résultats et des améliorations successives. Du point de vue du paramètre n , on peut considérer que le résultat de [10] apporte un gain d'un facteur $\log^2 n$ par rapport au résultat le

11. Le rôle des paramètres pour la caractérisation de la complexité des problèmes combinatoires fût profondément étudié dans [24]; par ailleurs, les liens entre la complexité paramétrique et l'approximation polynomiale sont étudiés (quoique superficiellement) dans [38].

plus trivial alors que du point de vue du paramètre Δ , les nombreux travaux relatifs à ce problème n'ont longtemps permis de n'apporter qu'une amélioration d'un facteur constant, l'enjeu de chaque amélioration étant d'améliorer la constante. Les résultats de la forme $k/\Delta - o(1/\Delta)$ pour tout k ([22]) ont clos cette question pour poser celle du dépassement de l'ordre de grandeur $O(1/\Delta)$; nous l'avons déjà évoquée dans le paragraphe 3.1. Une première réponse positive à cette question réside dans le théorème suivant, où nous notons WSTABLE le problème du stable pondéré (cf., annexe A).

Théorème 2. ([23]) *WSTABLE est approximable au niveau*

$$\min \left\{ \frac{\log n}{3(\Delta + 1) \log \log n}, O\left(n^{-4/5}\right) \right\} \quad (6)$$

Ce résultat est intéressant par sa forme: dans l'expression (6) le terme de droite est un rapport qu'on ne peut garantir en temps polynomial pour toute instance ([36]), ce qui rend le terme de gauche prépondérant. Or, celui-ci est justement une expression qui domine l'ordre de grandeur $O(1/\Delta)$. L'autre remarque intéressante sur ce théorème est qu'il fait conjointement intervenir les paramètres n et Δ . Enfin, notons que le résultat est directement établi pour le problème pondéré et que jusqu'à présent, aucune amélioration n'est connue pour le problème non-pondéré; nous aurons l'occasion d'y revenir à cette question dans la partie 6.

Cette question des paramètres se pose aussi pour COLORATION . On peut chercher à opposer au théorème 1 des résultats fonction du degré Δ . On peut notamment obtenir des énoncés du même type que ceux sur STABLE , ce qui montre l'intérêt de cette question qui n'est pas le fait d'un comportement exceptionnel du problème STABLE .

Théorème 3. ([23]) $\forall \epsilon$, *COLORATION est approximable au niveau*

$$\min \left\{ \frac{\log n}{\Delta \log \log n}, O\left(n^{-\epsilon}\right) \right\} \quad (7)$$

Pour les problèmes pondérés, les poids maximum et minimum constituent des paramètres intéressants. Dans de nombreux cas, notamment pour les problèmes STABLE et CLIQUE , les analyses d'algorithmes pour les versions pondérées peuvent aboutir à des rapports d'approximation indépendants des poids. Par exemple, le théorème 2 l'illustre. Cette situation est considérée comme favorable vu que les poids peuvent prendre de grandes valeurs, même pour des instances petites.

Pour d'autres problèmes par contre, les poids interviennent dans les analyses. Arrêtons-nous un instant sur le cas, très particulier, du problème de *stable maximal de poids minimum* (noté WSTABLEMAXMIN dans ce qui suit). Nous établissons le résultat suivant pour la version PWSTABLEMAXMIN correspondant au cas où les poids sont bornés par un polynôme; le résultat est donc a fortiori vrai pour WSTABLEMAXMIN . Remarquons en fait que le résultat reste vrai dès qu'on fixe une valeur de $W_{\max} > n$, ce qui servira dans la proposition 13 de la section 6.

Proposition 8. *Si $\mathbf{P} \neq \mathbf{NP}$, alors aucun algorithme ne peut garantir le rapport $n/(4w_{\max})$ pour PWSTABLEMAXMIN .*

Preuve. La démonstration se fait par réduction du problème de 3-COLORABILITÉ qui est $\mathbf{NP}$ -complet ([28]). Une instance est un graphe et la question est de savoir s'il peut être coloré avec trois couleurs de sorte que deux sommets adjacents n'ont pas la même couleur.

Soit $G = (V, E)$ une instance de 3-COLORABILITÉ. On construit alors le graphe pondéré $(G' = (V', E'), w)$, instance de WSTABLEMAXMIN , de la manière suivante:

$$\begin{cases} V' = V \times \{1, 2, 3, 4\} \\ ((v, i); (v', j)) \in E' \Leftrightarrow [(i = j) \wedge vv' \in E] \vee i \neq j \end{cases}$$

Le graphe G' est donc constitué de quatre copies de G , les copies de chaque sommet formant des cliques de taille 4. On affecte aux sommets des trois premières copies le poids 1 : pour $i = 1, 2, 3$, $W_{(v,i)} = 1$. On affecte aux sommets de la quatrième copie le même poids $W > n = |V|$. Ici $W = w_{\max}$; pour garantir au graphe pondéré construit d'être instance de PWSTABLEMAXMIN, il suffit de veiller à ce que $W \leq n^2$. Il est clair que la construction peut être effectuée en temps polynomial. En particulier, G' a $4n$ sommets.

Si G est 3-colorable, soit S_1, S_2, S_3 trois ensembles stables de G disjoints deux à deux et couvrant V . L'ensemble $S' = S_1 \times \{1\} \cup S_2 \times \{2\} \cup S_3 \times \{3\}$ constitue un stable de G' . De plus, il est maximal pour l'inclusion car pour tout sommet (v, i) de V' , soit $i \in \{1, 2, 3\}$ et $v \in S_i$ — dans ce cas $(v, i) \in S'$ — soit il existe $j \neq i$ tel que $v \in S_j$ et dans ce cas (v, j) est un voisin de (v, i) qui est dans S' . Par ailleurs, le poids de S' est $w(S') = |S_1| + |S_2| + |S_3| = n$. La valeur minimum d'un stable maximal est donc au plus n .

Si G n'est pas 3-colorable, considérons S' un stable maximal de G' et posons, pour $i \in \{1, 2, 3, 4\}$, $S'_i = S' \cap (V \times \{i\})$ et $S_i = \{v \in V, (v, i) \in S'_i\}$. Comme $S' = S'_1 \cup S'_2 \cup S'_3 \cup S'_4$ est stable dans G' , les ensembles S_1, S_2, S_3 et S_4 sont des stables disjoints de G . Par ailleurs comme G n'est pas 3-colorable, il existe $v \in V \setminus (S_1 \cup S_2 \cup S_3)$. L'ensemble $S'_1 \cup S'_2 \cup S'_3 \cup (v, 4)$ est alors un ensemble stable, ce qui signifie que $S'_1 \cup S'_2 \cup S'_3$ n'est pas maximal. Comme S' est maximal, $S_4 \neq \emptyset$ et alors $w(S') > W$.

Par conséquent, supposons qu'un algorithme polynomial pour WSTABLEMAXMIN garantisse le rapport $n/(4w_{\max})$ pour tout graphe d'ordre n avec pour poids maximum $w_{\max}$. Pour G' d'ordre $4n$ et de poids maximum W , le rapport garanti serait n/W . Par conséquent, si G est 3-colorable, la solution retournée par l'algorithme serait de valeur au plus W alors que dans le cas contraire elle serait de valeur strictement supérieure à W . En conclusion cet algorithme permettrait de décider, en temps polynomial, si G est 3-colorable ou non, ce qui conclut la preuve. ■

En particulier, il suffit de poser, pour $w_{\max}$, une valeur polynomiale en n pour interdire un rapport du type $P(1/n)$ où P est un polynôme. Ceci signifie que le problème PWSTABLEMAXMIN pour lequel les poids sont polynomialement bornés est hors de la classe **Poly-APX**. Ce résultat montre une différence notoire entre WSTABLEMAXMIN et WSTABLE, ce dernier problème (donc en particulier sa version PWSTABLE) appartiennent évidemment à **Poly-APX** : il suffit de sélectionner un stable contenant le poids maximum pour garantir le rapport $1/n$. Si on s'intéresse maintenant à la version générale de WSTABLEMAXMIN, remarquons que pour toute fonction $f(G)$ associant à un graphe non pondéré une valeur dans $]0, 1]$, aucun algorithme pour WSTABLEMAXMIN ne peut garantir le rapport $f(G)$. En d'autres termes, tout rapport d'approximation pour ce problème dépend nécessairement des poids. Il s'agit d'une situation tout à fait remarquable pour laquelle un paramètre (plus exactement une famille de paramètres) est incontournable.

Ce résultat illustre la diversité des situations pouvant intervenir dans le domaine des rapports non-constants et apporte une justification de plus à la question des paramètres et à leur prise en compte dans le formalisme de l'approximation. Cette question rejoint souvent la notion d'ordre de difficulté comme l'illustre l'exemple de CLIQUE. Le but est d'ailleurs le même, à savoir comprendre comment les différents paramètres de l'instance conditionnent les possibilités de résolution. Le cadre des réductions abordé dans la partie 6 fera ressurgir cette problématique.

5.2 Chaînes d'approximation

La possibilité d'étudier globalement le comportement d'une famille d'algorithmes constitue l'un des apports de ce travail. Il s'agit également d'une forme de paramétrage, par rapport à l'algorithme cette fois.

L'ensemble des fonctions $f \in \mathcal{F}_\Pi$ constantes en I , *i.e.*, des fonctions pour lesquelles il existe une suite $(x_k)_{k \in \mathbb{N}}$ de $]0, 1[$ telle que $\forall(I, k) \in \mathcal{I} \times \mathbb{N}, f(I, k) = x_k$, définit le niveau des *chaînes indépendantes de l'instance*. Remarquons que le cas où (x_k) est une suite convergeant vers 1 correspond à un schéma d'approximation polynomial. Cette notion se distinguait jusqu'ici parmi les outils d'approximation. Le formalisme unifié permet de la rapprocher d'autres situations et de définir la classe **PTAS** comme un niveau d'approximation.

La convergence vers 1 a un statut particulier puisque que le rapport limite correspond à la résolution exacte. La première extension est la convergence d'une chaîne vers un autre rapport constant. Afin d'illustrer cette situation, citons un résultat pour le problème H-3-TRANSVERSAL correspondant aux instances de H-TRANSVERSAL pour lesquelles les ensembles contiennent au plus 3 éléments. Un résultat établi dans [32] s'interprète comme une convergence.

Théorème 4. ([32]) *H-3-TRANSVERSAL admet une chaîne d'approximation polynomiale à rapport (γ) convergeant vers $5/7$.*

Le cadre de l'approximation de STABLE en fonction du paramètre Δ fut la première motivation à considérer des chaînes d'approximation. Nous reformulons ci-dessous certains résultats évoqués dans le paragraphe 3.1 afin de mieux les comparer et de montrer comment le formalisme que nous avons introduit permet de simplifier leur énoncé.

Proposition 9.

- 1° STABLE admet une chaîne d'approximation polynomiale de complexité $O(n^k)$ convergeant asymptotiquement (par rapport à l'ordre de difficulté Δ) vers $(6/\Delta)$ ([34]).
- 2° Il existe une chaîne d'approximation polynomiale de complexité $O(n^{k/2})$ pour STABLE garantissant le rapport d'approximation asymptotique (par rapport à Δ) k/Δ ([22]).
- 3° Il existe une chaîne d'approximation polynomiale de complexité $O(n^k)$ pour STABLE garantissant le rapport d'approximation asymptotique (par rapport à μ)

$$\min \left\{ \frac{k}{\mu}, k' \frac{\log \log \Delta}{\Delta} \right\} \quad (8)$$

Le résultat du point 1 est plus compréhensible que sous sa forme originelle. Par ailleurs, cette présentation permet de le comparer au résultat du point 2 : pour $k = 6$, celui-ci correspond à un algorithme de complexité $O(n^3)$ garantissant le rapport asymptotique $6/\Delta$. Le gain par rapport au point 1 réside donc, d'une part dans la complexité, et d'autre part dans le type de garantie : un rapport asymptotique contre une convergence asymptotique.

Bien entendu, ces résultats ont été supplantés par le théorème 2. Cependant, ce dernier ainsi que le théorème 3 (expression (7)) ont une version en termes de chaînes. Dans le théorème 5, les termes fonction de Δ sont moins bons que leur équivalent dans les théorèmes 2 et 3 ; par contre, les termes auxquels ils sont opposés sont encore plus improbables que dans ces deux théorèmes.

Théorème 5. ([23])

- 1° WSTABLE admet une chaîne d'approximation garantissant comme rapport le minimum entre $k/(\Delta + 1)$ et $O(\log^{-3(k+1)/4} n)$;
- 2° $\exists k' > 0$, tel que COLORATION admet une chaîne d'approximation garantissant comme rapport (γ) le minimum entre $k'k/\Delta$ et $k' \log^{-3k/4} n$.

5.3 Approximation différentielle

Le cadre du rapport différentiel δ donne lieu à différents types de résultats. Nous en citons ici quelques uns pour mémoire.

Un résultat assez remarquable est que COLORATION, particulièrement difficile à approximer du point de vue classique (γ) , se prête bien à l'approximation différentielle. En effet, il est

dans ce cadre approximable à rapport constant : les différents rapports différentiels obtenus ont été $1/2$ ([17]), $2/3$ ([35]), $5/7$ et $3/4$ ([32]) et enfin $289/360$ ([25]) ; par contre, si $\mathbf{P} \neq \mathbf{NP}$, il n'admet pas de schéma d'approximation polynomial ([32]).

Le problème de BINPACKING, de son côté, a donné lieu à plusieurs analyses différentielles. Dans [19] par exemple, nous analysons les différents algorithmes gloutons pour ce problème. Parmi eux, citons par exemple l'algorithme FFD (*first-fit-decreasing*) qui consiste à prendre les nombres par ordre décroissant et à les placer dans la première boîte pouvant les contenir. Du point de vue du rapport classique, cet algorithme garantit le rapport $2/3$ ([57]). Du point de vue différentiel, nous montrons qu'il garantit le rapport $3/4$, cette borne étant atteinte. Par contre, son analyse asymptotique met en évidence le rapport $7/9$ par rapport à l'ordre de difficulté $\sigma(L)$ (le support de l'instance défini au paragraphe 4.2.3). Notons d'ailleurs que la notion de support nous a initialement été utile pour définir un rapport différentiel asymptotique. Ici, $\sigma(L)$ est un ordre de difficulté par rapport à la résolution exacte comme la proposition 19 nous permettra de le montrer. D'après la proposition 6, FFD peut donc être facilement converti en une chaîne d'approximation convergeant vers $7/9$. Plus généralement, l'étude différentielle des différents algorithmes gloutons pour BINPACKING mettent en évidence des phénomènes relativement similaires au cas classique.

Par contre, du point de vue général de l'approximation, BINPACKING s'avère mieux approximable sous le rapport différentiel que sous le rapport classique. En effet, nous obtenons dans [18] un schéma d'approximation différentiel qu'on ne peut obtenir du point de vue classique (proposition 1).

6 Comparer les problèmes : un complément pour structurer NPO

6.1 Motivations

Nous avons déjà mentionné l'enjeu de l'approximation pour définir une structure des problèmes **NPO** en fonction de leur approximabilité. La notion de niveau d'approximation est une première manière d'établir une telle hiérarchie. La combinaison de résultats positifs et négatifs positionne les problèmes dans cette structure. Cependant, cette classification n'est pas totalement satisfaisante pour deux raisons.

La première est que cette hiérarchie n'est pas pérenne en ce sens qu'elle est amenée à évoluer en fonction de nouveaux résultats. En effet, il reste souvent une marge importante entre le meilleur résultat d'approximation et le meilleur seuil de difficulté connus pour un problème donné. Par conséquent, le niveau d'approximation auquel appartient le problème sera amené à évoluer en fonction de prochains résultats. De même, il est évident que les résultats d'approximation relatifs à un algorithme ne permettent pas de comparer différents algorithmes tant que les analyses ne sont pas prouvées optimales. C'est ainsi, par exemple, que l'algorithme glouton pour STABLE a été redécouvert en 1994 comme garantissant le rapport $O(3/\Delta)$; il supplanta alors les différents algorithmes garantissant $O(2/\Delta)$ qui avaient été, à leur époque, présentés comme des améliorations de l'algorithme glouton.

La seconde raison, plus profonde, est que les analyses d'approximation, même lorsqu'elles sont optimales, sont parfaitement adaptées pour comparer différents résultats relatifs à un même problème. Par contre, il n'est pas toujours pertinent de comparer directement des résultats d'approximation pour des problèmes différents, surtout lorsque les approximations mises en évidence sont du même type. Par exemple, comparer la valeur des rapports d'approximation de deux problèmes de **APX** n'a souvent que peu de sens. De même, deux problèmes identifiés à un même niveau d'approximabilité n'ont pas nécessairement de lien entre eux.

Ces remarques ne sont en rien une critique des résultats d'approximation. Elles doivent simplement être mises en évidence pour qu'il ne soit pas fait un mauvais usage de telles analyses. Une première réponse est qu'enrichir les outils et les points de vue est un premier palliatif à ces

imperfections. Ainsi, les analyses différentielles nous ont convaincu que regarder les problèmes sous deux (ou plus) angles différents est riche d'enseignements. Par exemple, COLORATION, particulièrement difficile à analyser du point de vue du rapport classique, est tout à fait adapté à des analyses différentielles. La différence de comportement entre ce problème et H-TRANSVERSAL est inversée lorsqu'on passe du cadre classique au cadre différentiel, ce qui permet de relativiser les conclusions hâtives que l'on pourrait tirer en les comparant sous un seul des deux points de vue. De même, analyser la difficulté des problèmes par rapport à différents paramètres permet de compenser certains biais d'une analyse fondée exclusivement sur la taille de l'instance. L'exemple de la comparaison de CLIQUE et de STABLE est significatif. Ceci justifie qu'on cherche à établir un formalisme unifié permettant de prendre en compte, systématiquement et au même niveau, ces différents points de vue complémentaires. Cette richesse de problématiques et de concepts n'a pas qu'un rôle de garde-fou contre des conclusions trop hâtives ; elle permet réellement une meilleure compréhension des problèmes.

Ce constat des limites des outils d'analyse d'algorithmes pour concevoir une structure de la classe **NPO** justifie la mise au point d'outils spécifiques permettant de comparer la difficulté d'approximation de différents problèmes. C'est l'objet des *réductions en approximation* auxquelles nous nous intéressons dans cette section.

L'idée est de concevoir un préordre sur **NPO** permettant de confronter la difficulté d'approximation de différents problèmes. L'intérêt est alors d'obtenir une structure des problèmes qui représente leurs liens intrinsèques indépendamment du niveau de connaissance que l'on a de chacun d'eux. La structure générale issue des travaux d'approximation est la combinaison de résultats d'approximation, qui permettent de situer les problèmes sur une échelle absolue, et de résultats comparant directement différents problèmes du point de vue de l'approximation.

Ces deux types d'études sont complémentaires et intimement liés. En particulier, il est important de souligner que les réductions permettent de transférer des résultats d'approximation d'un problème à un autre. Cet intérêt opérationnel est réel : de très nombreux résultats d'approximation actuels sont issus de tels transferts. Par ailleurs, une réduction entre deux problèmes permet d'étalonner les échelles de qualité pour ceux-ci et de donner un sens à la comparaison directe des résultats d'approximation les concernant. Mentionnons enfin que comme tout préordre, les réductions offrent une notion d'équivalence qui constitue un nouveau moyen de concevoir des classes de problèmes. En particulier, les classes de complétude (classe de majorants) jouent un rôle important.

Remarquons enfin que ce type d'outil est à la base de toutes les démarches visant à envisager une notion de difficulté d'un problème, aussi bien dans le cadre de la théorie de la décidabilité que dans celui de la complexité. En particulier, les démonstrations de **NP**-complétude ([28]) reposent sur la notion de réduction polynomiale. Tous les problèmes d'optimisation que nous envisageons ont une version décision **NP**-complète. Leurs versions décision sont donc mutuellement liées par des réductions polynomiales. La diversité des comportements par rapport à l'approximation montre que de telles réductions ne préservent pas les propriétés d'approximation. En fait, une réduction polynomiale entre les versions décision de deux problèmes **NPO** correspond, pour les versions optimisation, à une transformation qui préserve les solutions optimales. Les réductions en approximation sont des cas particuliers de réductions polynomiales préservant, non seulement les solutions optimales, mais aussi les « bonnes solutions ».

6.2 Généalogie

Nous proposons dans ce paragraphe un rapide aperçu de l'évolution des concepts de réduction dans la continuité des *réductions algorithmiques*. Ces dernières années, l'approximation polynomiale a connu une profusion de définitions de réductions de haute technicité et il est assez difficile

de s'orienter dans ce « bestiaire ». Cet historique n'a pas vocation à les décrire toutes ni à en donner une définition précise. Notre but est de montrer une certaine logique dans cette évolution afin d'introduire, dans le paragraphe 6.3, une nouvelle notion de réduction, notée FP, qui unifie, sous un certain point de vue, l'ensemble des réductions existantes. Nous insistons sur les restrictions permettant cette unification car elles conditionnent l'usage que l'on peut faire des réductions FP. Nous donnons également des pistes pour des généralisations ultérieures. Dans la section 6.5 nous proposons des exemples d'utilisation de ces concepts avant de les exploiter, dans la section 7, pour comprendre la structure des instances d'un problème.

Étant donnés deux problèmes de décision Π_1 et Π_2 , une *réduction algorithmique* f de Π_1 à Π_2 associe (algorithmiquement) à toute instance du premier une instance du second ayant la même réponse (vrai ou faux). La réduction permet de transformer tout algorithme A pour Π_2 en un algorithme pour Π_1 par simple composition ($A \circ f$) suivant le schéma suivant :

- (1) utiliser la réduction pour construire une instance de Π_2 ;
- (2) lui appliquer l'algorithme A ;
- (3) retourner la même réponse.

Le problème Π_2 est alors au moins aussi difficile que Π_1 . Dans le contexte de la décidabilité, f a juste besoin d'être TURING-calculable ; la réduction polynomiale ([43]) suppose en outre une hypothèse de complexité. Le schéma précédent permet alors de transformer tout algorithme polynomial pour Π_2 en un algorithme polynomial pour Π_1 .

Dans le cadre de l'approximation, dès 1974 JOHNSON évoquait (en conclusion de [40]) la possibilité de réductions plus fines permettant de transférer des résultats d'approximation. Les premiers exemples sont apparus vers 1977 ([5]) puis se sont considérablement développés pour conduire, aujourd'hui, à une dizaine de notions de réductions, notamment L, LINÉAIRE¹², CONTINUE, STRICTE APX, PTAS, E, S, AP (un bref descriptif formel de ces réductions peut-être trouvé dans [14]).

Les réductions polynomiales ont d'abord été affinées de manière à préserver la « structure » des instances ([8, 48, 5]) avant que ne soit mis en évidence, dans [5], leur capacité à préserver l'approximation. Conformément au schéma d'origine, elles sont toutes conçues sur ce même modèle (que nous appelons *d'instance à instance*, notée l-l) consistant en une transformation polynomiale f de toute instance I_1 de Π_1 en une instance $f(I_1) = I_2$ de Π_2 . Dans le cadre **NPO**, l'égalité des réponses pour les deux instances n'a plus lieu d'être ; il faut donc enrichir la réduction par un algorithme transformant une solution approchée (réalisable) pour I_2 en une pour I_1 . Cet algorithme pouvant lui-même dépendre de I_1 ; nous considérons un algorithme polynomial h associant à toute instance I_1 , l'algorithme (polynomial) $h_{I_1} : \mathcal{C}_{I_2} \rightarrow \mathcal{C}_{I_1}$. La composition $h \circ A \circ f$ ¹³, où A est un algorithme approché pour Π_2 , fournit alors un algorithme approché pour Π_1 .

Les différentes réductions en approximation se distinguent par des hypothèses permettant de comparer la qualité des solutions y et $h_{I_1}(y)$. Dans certains cas, comme pour les réductions continues ([58]) et linéaires ([54]), ces hypothèses sont directement formulées sous forme de bornes liant, d'une part les valeurs optimales de I_1 et $f(I_1)$ et, d'autre part les valeurs objectives de y et $h_{I_1}(y)$. Pour d'autres, comme pour la réduction STRICTE ([52]) et la réduction E ([44]), la définition explicite le rapport d'approximation de $h_{I_1}(y)$ en fonction du rapport pour y . Enfin (notamment pour les réductions A ([52]), P ([16]), APX, PTAS ([4]) ou encore AP ([15])), le lien direct entre les rapports garantis par y et $h_{I_1}(y)$ est remplacé par une implication du type : « si le rapport pour y a telle garantie, alors celui pour $h_{I_1}(y)$ aura telle autre garantie ».

Cette classification met en évidence une évolution au cours de laquelle les liens directs entre les instances I_1 et $f(I_1)$ (hérités des réductions polynomiales) sont partiellement oubliés au

12. À ne pas confondre avec l'équivalence linéaire de [13].

13. Nous désignons ainsi $I \mapsto h_I \circ A \circ f(I)$.

profit du comportement de la réduction vis à vis de l'approximation : quelles garanties offre l'algorithme $h \circ A \circ f$ par rapport à celles de A ? Les réductions continues par exemple, sont connues pour préserver les rapports constants alors que les réductions linéaires ne préservent que les schémas d'approximation polynomiale. La réduction **E** est conçue pour être compatible avec des classes d'approximation moins restrictives. Cependant, elles butent toutes sur une même limite : elles n'imposent, dans le meilleurs des cas, que des liens entre les valeurs des rapports pour y et $h_{I_1}(y)$ sans comparer l'ordre des différents paramètres des instances I_1 et $f(I_1)$. Elles se comportent donc en général très bien au sein du niveau d'approximation **APX** mais permettent difficilement de rendre compte du transfert de rapports dépendant de l'instance. Tout au mieux, la réduction **E** préserve, dans certaines conditions, les rapports logarithmiques, polynomiaux et de la forme $1/o(n^{1-\epsilon})$, mais pas ceux de la forme $1/n^{1-\epsilon}$. La notion d'*amplification* introduite par Kann ([41]) tente d'y remédier en spécifiant le lien entre les tailles de I_1 et $f(I_1)$, mais son exploitation est restée jusqu'à présent très limitée.

6.3 Réduction à expansion fonctionnelle FP

Cette explosion du nombre de définitions et leur niveau technique rend souvent leur exploitation difficile. C'est pourquoi nous proposons une nouvelle réduction appelée **FP** qui permet d'unifier la plupart des notions existantes tant qu'on se limite au critère du comportement par rapport à l'approximation.

On reste dans un schéma $l-l$; par contre, on oublie tout lien direct entre les couples $(I_1, h_{I_1}(y))$ et $(f(I_1), y)$ en se focalisant sur la correspondance des niveaux d'approximation garantis par les algorithmes A (pour Π_2) et $h \circ A \circ f$ (pour Π_1). L'*expansion* exprime directement ce lien au moyen d'une application g opérant sur un ensemble de rapports d'approximation. Le terme est emprunté à la définition de réduction **CONTINUE** ([58]) pour laquelle la transformation est une simple homothétie dont le rapport est appelé expansion. Ici, la transformation peut être quelconque ; nous parlons d'*expansion fonctionnelle* et qualifions ces réductions de **FP**. Jusqu'ici, les rares notions d'expansion se limitaient à des fonctions réelles d'une forme spécifique dans le cadre **APX**. Une motivation importante dans notre démarche est la capacité à prendre en compte des niveaux d'approximation au delà de **APX**.

Définition 7. Soient $(\Pi_1, \Pi_2) \in \mathbf{NPO} \times \mathbf{NPO}$. Une réduction **FP** de Π_1 à Π_2 est un triplet $\propto_{\Pi_1, \Pi_2} = (f, h, g)$ tel que :

- 1° $f : \mathcal{I}_1 \times \mathbb{N} \rightarrow \mathcal{I}_2 \times \mathbb{N} : (I_1, k_1) \mapsto f(I_1, k_1) = (I_2, k_2)$ est de complexité polynomiale ;
- 2° $h : (I_1, k_1) \in \mathcal{I}_1 \times \mathbb{N} \mapsto [h_{I_1, k_1} \in \mathcal{C}_{I_1}^{I_2}], \forall x_2 \in \mathcal{C}_{I_2}, \forall h_{I_1, k_1}(x_2) \in \mathcal{C}_{I_1}$ est calculable en temps polynomial par rapport à $|I_1|$;
- 3° $g : \mathcal{F}_{\Pi_2} \rightarrow \mathcal{F}_{\Pi_1}$;

de sorte que pour toute chaîne d'approximation $(A_k^2)_{k \in \mathbb{N}}$ pour Π_2 , garantissant le rapport ρ_2 , $h \circ (A_k^2) \circ f : (I_1, k_1) \mapsto h_{I_1, k_1}(A_{k_2}^2(I_2))$ est une chaîne garantissant $g(\rho_2)$. ■

L'expansion représente l'effet de la transformation sur la qualité d'approximation. Une réduction permet de comparer les niveaux d'approximation des problèmes Π_1 et Π_2 . Pour deux niveaux $F_1 \subset \mathcal{F}_{\Pi_1}$ et $F_2 \subset \mathcal{F}_{\Pi_2}$:

- si $g(F_2) \subset F_1$, alors on dit que la réduction $\propto_{\Pi_1, \Pi_2}$ transforme le niveau F_2 en F_1 ;
- si $g(F_2) \subseteq F_2$ et si les deux problèmes ont le même ensemble d'instances ou lorsque la définition de F_1 et F_2 est indépendante des instances, on dira que $\propto_{\Pi_1, \Pi_2}$ préserve le niveau F_2 .

La notion est la même que l'on considère le rapport classique ou le rapport différentiel ; on peut même imaginer (paragraphe 6.5) des réductions entre les deux points de vue (il suffit d'inclure dans g la nature des rapports (γ ou δ) au départ et à l'arrivée).

Dans certains cas, on ne parvient à transformer polynomialement que certaines chaînes d'approximation pour Π_2 en une chaîne d'approximation pour Π_1 , on parle alors de *réduction partielle* dont l'usage est restreint à deux niveaux F_1 et F_2 .

L'intérêt de cette notion est double. D'une part elle permet d'étendre les concepts classiques sans restriction a priori du type d'approximation que l'on peut transférer : la réduction FP peut s'adapter à tous les types de transfert alors que chaque notion classique est d'usage restreint. Par ailleurs, la plupart des réductions que nous avons évoquées peuvent s'interpréter comme des cas particuliers de réduction FP avec une forme spécifique pour l'expansion.

Ainsi, par exemple :

- la L-réduction ([54]) est un cas particulier de réduction partielle préservant **PTAS** (*i.e.*, le niveau des chaînes convergeant vers 1) ;
- la *réduction faiblement continue* étudiée par SIMON dans [58] devient un cas particulier de réduction partielle préservant **APX** (algorithmes à rapport constant) ; l'expansion est alors une homothétie de rapport constant ; en appliquant la réduction à chaque algorithme d'une chaîne, cette réduction permet de préserver les chaînes à rapport convergeant dans **APX** ;
- la réduction E ([44]) est conçue pour préserver, non seulement **APX** et **PTAS**, mais aussi certains niveaux d'approximation faible, notamment **Log-APX** et **Poly-APX**, mais pas des niveaux associés à un polynôme de degré fixé ; dans ce cas, l'expansion est de la forme $g : \rho \mapsto (1 - (\alpha(1 - \rho)/(\rho + \alpha(1 - \rho)))) \circ f$ où α est une constante ;
- les réductions PTAS et AP ([15]) sont des cas très particuliers car elles dépendent du rapport d'approximation ; elles ne permettent de transférer sur Π_1 que des algorithmes garantissant un rapport déterminé pour Π_2 .

Bien entendu, on ne peut pas espérer caractériser totalement les réductions usuelles en termes de réduction FP vu que cette dernière a pour but d'oublier les liens structurels entre les instances I_1 et I_2 , liens sur lesquels reposent toutes les autres définitions. La réduction FP est en quelque sorte une projection des notions classiques sur le critère unique du transfert d'approximation. Le comportement par rapport à l'approximation est inclus dans la définition (via l'expansion) alors que pour toutes les autres notions il est une conséquence de la définition. Les définitions classiques restent primordiales pour certaines études structurelles. En particulier la réduction E a joué un rôle important pour lier les classes d'approximation à des classes *syntaxiques* définies par une écriture logique de la requête associée au problème. Par contre, la définition que nous proposons apporte une grande simplification et des possibilités beaucoup plus larges dès lors qu'on ne s'intéresse qu'aux propriétés d'approximation.

6.4 Réductions du troisième type

Le cadre que nous venons de décrire reste assez restrictif quant à la forme d'instance à instance (réduction l-l) de la réduction FP. Ce carquois est utile pour certaines études, notamment pour les notions de complétude en approximation. L'autre intérêt était d'unifier, avec le moins de restrictions possible, les notions usuelles.

Par contre, en poursuivant la logique du transfert des propriétés d'approximation, on peut imaginer des situations beaucoup plus générales où, pour résoudre une instance de Π_1 , on a besoin de résoudre, non pas une mais plusieurs (nombre polynomial) instances de Π_2 . On parle alors de réduction *d'instance à problème*, notée réduction l-P ; toutes les notions précédentes, notamment celle d'expansion, peuvent se généraliser à ce cas. Certaines problématiques nécessitent de faire cette distinction, notamment lorsqu'on cherche à comparer la difficulté relative entre approcher la valeur optimale du problème et déterminer une solution approchée. En pratique, de très nombreuses réductions sont de la forme l-P, d'où l'importance d'intégrer ce cas au formalisme.

Citons par exemple une démarche très naturelle pour colorer un graphe en exploitant un algorithme pour STABLE. La première couleur est composée de l'ensemble stable obtenu en appliquant l'algorithme au graphe complet. On efface alors les sommets déjà colorés ainsi que les arêtes incidentes et on réitère le processus. L'idée sous-jacente est que prendre à chaque fois un stable le plus grand possible est compatible avec l'objectif de couvrir les sommets avec un minimum de couleurs. La réduction qui en résulte est de type l-P. Nous analysons son expansion dans la prochaine partie.

6.5 Quelques exemples

Nous développons dans cette partie des exemples de réductions qui mettent en évidence l'intérêt des notions et des problématiques que nous avons exposées. Ils illustrent aussi l'usage des réductions pour étudier, non seulement des liens entre deux problèmes fixés, mais aussi pour envisager des problématiques générales telles que le rôle des poids dans un problème pondéré, le rôle des paramètres, les liens entre rapports classique et différentiel, ...

6.5.1 Quelques grands classiques

Le fait d'avoir choisi, pour la réduction FP, le cadre d'instance à instance l-l permet de garantir qu'une réduction FP préservant la résolution exacte entre deux problèmes NPO correspond à une réduction polynomiale entre les versions décision de ces problèmes. Par ailleurs, un résultat de [55] peut s'interpréter comme la réciproque, sous hypothèse de complétude : une réduction polynomiale entre les versions décision NP-complètes de deux problèmes NPO permet d'établir une réduction FP préservant la résolution exacte entre ces problèmes.

Il est de même très facile de réécrire des résultats négatifs tels que les propositions 1 et 8 comme des réductions FP transformant un certain niveau d'approximation pour un problème en une résolution exacte pour un autre.

6.5.2 Stratégie maître-esclave

Revenons à l'exemple du lien entre COLORATION et STABLE que nous avons choisi pour illustrer les réductions l-P. Il s'inscrit dans le schéma général *maître-esclave* décrit dans [40, 58]. Le problème maître est un problème de minimisation (COLORATION dans notre cas) consistant à partitionner un ensemble de base en un nombre minimum d'ensembles vérifiant une propriété π . Le problème esclave consiste en la maximisation d'une partie vérifiant π . On s'intéresse alors au schéma de résolution du problème maître en faisant appel de manière itérative au problème esclave. Il s'agit en fait d'une réduction l-P du maître à l'esclave. Dans [40], il est montré qu'elle transforme la résolution exacte en un rapport $1/\log n$ où n est la taille de l'ensemble de base. Ce résultat est étendu dans [58] : un rapport constant ρ pour le problème esclave est transformé en $\rho/\log n$. Ce problème est revisité dans [1]; le résultat s'étend aux rapports dépendant de l'instance.

La *coloration d'arêtes*, quant à elle, consiste à colorer les arêtes d'un graphe en un nombre minimum de couleurs de sorte que deux arêtes adjacentes n'ont pas la même couleur. Dans ce cas, le problème esclave est le couplage maximum qui est polynomial ; il en résulte le rapport $1/\log n$ pour le problème de coloration d'arêtes.

6.5.3 Liens entre versions pondérée et non-pondérée

6.5.3.1 Cas du stable maximum

Considérons d'abord WSTABLE pour lequel des poids sont affectés aux sommets du graphe. Dans le cas de poids positifs entiers, une construction proposée par SIMON ([58]) permet de

formuler WSTABLE dans un graphe G comme la recherche d'un stable de cardinal maximum dans un graphe construit à partir de (G, w) noté G_w . L'idée est la suivante : chaque sommet v_i de poids w_i dans G donne lieu à w_i sommets $v_{i,l}$, $l = 1, \dots, w_i$ dans G_w qui forment un stable. Une arête $v_i v_j$ de G donne lieu, dans G_w , à toutes les arêtes $v_{i,l} v_{j,k}$ avec $l = 1, \dots, w_i$, $k = 1, \dots, w_j$, c'est-à-dire à un graphe biparti complet. Tous les sommets $v_{i,l}$, $l = 1, \dots, w_i$ issus d'un même v_i ont donc le même voisinage dans G_w de sorte qu'un stable maximal de ce graphe contient toutes les copies de v_i ou aucune. Par conséquent, tout stable S de G de poids $w(S)$ correspond à un stable de cardinal $w(S)$ dans G_w composé de toutes les copies des sommets de S . Réciproquement, tout stable maximal S_w de G_w se projette en un stable de poids $|S_w|$ dans G . Ainsi, toute solution réalisable (resp., optimale) pour le problème WSTABLE dans G correspond à une solution réalisable (resp., optimale) pour STABLE dans G_w et réciproquement. Étant donnée la définition des problèmes WSTABLE et STABLE, la valeur objective est conservée par cette transformation. La seule restriction à laquelle il faut rester attentif est que la transformation de (G, w) en G_w n'est polynomiale que si les poids de l'instance (G, w) de WSTABLE sont bornés par un polynôme de la taille de G (PWSTABLE). En effet, l'ordre de G_w est exactement la somme des poids des sommets de G . Enfin, notons que le degré maximum des sommets de G_w est majoré par $w_{\max} \Delta$. La discussion ci-dessus donne lieu au résultat suivant.

Proposition 10. *Il existe une réduction partielle de PWSTABLE à STABLE transformant toute chaîne de rapport $f(\Delta, n, k)$ (f étant croissant en k et décroissant en Δ et en n) pour STABLE en une chaîne de rapport $f(w_{\max} \Delta, w(V), k)$ pour PWSTABLE.*

Initialement ([58]), cette réduction n'a été envisagée que pour des rapports indépendants de l'instance. Il est clair qu'elle préserve **APX** (la valeur du rapport est conservée), par contre, dès qu'on envisage des rapports dépendant de l'instance elle a le défaut d'induire des expressions fonction du système de poids. Ceci est dû au fait que les paramètres n et Δ pour le graphe G_w auquel on applique un algorithme pour STABLE dépendent des valeurs des poids. Cet exemple illustre tout à fait le problème qui se pose dès qu'on considère des rapports dépendant de l'instance. Au cours d'une réduction de Π_1 à Π_2 , le rapport d'approximation dépend de l'instance $f(I_1)$ à laquelle on applique un algorithme pour le problème Π_2 . Par contre, il faut l'exprimer par rapport aux paramètres de I_1 de sorte que la qualité de la réduction est tributaire de la valeur des paramètres de l'instance $f(I_1)$ par rapport à ceux de l'instance I_1 .

Dans le cas présent, ce problème est crucial puisque les algorithmes à envisager ne peuvent garantir que des rapports non-constants. Comme le problème pondéré WSTABLE admet des rapports indépendants des poids, cette réduction n'a d'intérêt que pour la version BWSTABLE correspondant au cas de poids bornés. Dans ce cas, elle préserve les chaînes à rapport (asymptotique ou non-asymptotique) de la forme k/Δ ou k/μ . Ainsi, les résultats de la proposition 9 (cf., expression (8)) s'appliquent à la version BWSTABLE.

Pour les problèmes STABLE et WSTABLE par contre, la question d'une réduction dont l'expansion ne dépend pas des poids se pose. Nous montrons, dans [23], le résultat suivant

Proposition 11. *Pour tout k , il existe une réduction de WSTABLE à STABLE qui transforme un rapport $\rho(G)$ en $\min\{\log^k n/2n, 0,099\rho(G)/(k \log \log n)\}$.*

C'est sur cette proposition que repose l'approximation de WSTABLE en $O(\log^2 n/n \log \log n)$ (paragraphe 3.1).

6.5.3.2 Cas du stable maximal minimum

La réduction qui associe le graphe G_w à un graphe pondéré (G, w) s'applique également aux problèmes STABLEMAXMIN et WSTABLEMAXMIN. L'analyse est essentiellement la même.

Proposition 12. *Il existe une réduction partielle entre PWSTABLEMAXMIN et STABLEMAXMIN transformant toute chaîne de rapport $f(\Delta, n, k)$ (f étant croissant en k et décroissant en Δ et en*

n) pour STABLEMAXMIN en une chaîne de rapport $f(w_{\max}\Delta, w(V), k)$ pour PWSTABLEMAXMIN. Elle a dans ce cas un intérêt plus grand puisque, pour le problème WSTABLEMAXMIN, tout rapport garanti en temps polynomial dépend nécessairement des poids (proposition 8). Son second intérêt est de donner la possibilité de récupérer un résultat négatif pour STABLEMAXMIN à partir de la proposition 8. Il en résulte un résultat négatif (proposition 13) pour le niveau $O(1/n^{1-\epsilon})$. Celui-ci a déjà été obtenu dans [29]; cependant, la preuve résultant de la concaténation des propositions 8, 12, et 13 est beaucoup plus simple. Nous mentionnons ce résultat et sa preuve car il montre un exemple simple de seuil de difficulté justifiant qu'on prenne en compte des subdivisions de **Poly-APX** (voir paragraphe 5.1.3). Il montre aussi que la proposition 8, qui semble être fortement liée aux valeurs des poids, porte en elle également une information d'ordre structural puisqu'elle est à la base d'un résultat négatif pour le cas non-pondéré. Enfin, il illustre l'intérêt qu'il peut y avoir à étudier, par le biais de réductions, la question des liens entre différentes versions d'un problème.

Proposition 13. *Si $P \neq NP$, pour tout $\epsilon \in]0, 1[$, STABLEMAXMIN n'est pas approximable au niveau $O(1/n^{1-\epsilon})$.*

Preuve. Considérons un algorithme polynomial garantissant un rapport du type $O(1/n^{1-\epsilon})$ avec $\epsilon \in]0, 1[$. Appliquons la réduction de la proposition 12 à un graphe pondéré à n sommets instance de PWSTABLEMAXMIN vérifiant $w_{\max} > \max\{n^{2/\epsilon-1}/4^{1/\epsilon}, n\}$. Il en résulte, pour de tels graphes, un algorithme polynomial garantissant le rapport $1/(nw_{\max})^{1-\epsilon} > 1/4w_{\max}$. La preuve de la proposition 8 montre comment un tel algorithme permettrait de résoudre la 3-colorabilité. ■

6.5.4 Sous graphe induit k -colorable

Toujours dans le but d'illustrer la notion de réduction FP et l'intérêt de considérer la question du transfert des paramètres par de telles transformations, intéressons-nous au problème kCl consistant à déterminer, dans un graphe, un sous-graphe k -colorable d'ordre maximum où k est une constante fixée. Ce problème est une généralisation de STABLE correspondant à $k = 1$. On pourrait imaginer, en suivant l'idée d'une démarche maître-esclave, une réduction l-P d'instance à problème de kCl à STABLE. Cependant, nous montrons ici que ces problèmes sont directement liés par une réduction FP du type l-l. L'idée est exactement la même que celle développée dans la proposition 8; cet exemple nous montre donc un autre cas où, comme pour les réductions des versions pondérée à non-pondéré (construction G_w), une même construction peut être exploitée pour les problèmes STABLE et STABLEMAXMIN.

Proposition 14. *Pour k fixé, il existe une réduction FP de kCl à STABLE (resp., de $WkCl$ à WSTABLE) transformant tout niveau d'approximation $f(n, \Delta)$ pour STABLE en $f(kn, \Delta + k - 1)$.*

Preuve. La réduction reprend la construction établie dans la preuve de la proposition 8 en la généralisant de 4 à k copies. Soit G une instance de kCl , on construit le graphe ${}^kG = (V_k, E_k)$ instance de STABLE de la manière suivante :

$$\begin{cases} V_k = V \times \{1, \dots, k\} \\ ((v, i); (v', j)) \in E_k \Leftrightarrow [(i = j) \wedge vv' \in E] \vee i \neq j \end{cases}$$

Il s'agit donc de k copies de G , les copies d'un même sommet constituant une clique de taille k . Pour le cas pondéré, on affecte, dans le graphe kG , le poids w_v à tous les sommets (v, i) , $i \in \{1, \dots, k\}$.

Tout stable maximal S de kG correspond dans G à un sous-graphe k -colorable maximal de même valeur. Les k stables constituant ce sous-graphe correspondent à la trace de S sur les k copies de G . La preuve est conclue en remarquant que kG est d'ordre nk et de degré $\Delta + k - 1$ si G est d'ordre n et de degré Δ . ■

Cette réduction préserve en particulier les chaînes à rapport (asymptotique ou non) ρ/Δ , ce qui permet de récupérer, pour kCl , les résultats de la proposition 9. Mais elle permet également de transférer à kCl les résultats d'approximation des théorèmes 1 et 2.

Proposition 15. *kCl est approximable au niveau $O(\log^2 n/n)$ et $WkCl$ est approximable au niveau $O(\min\{\log n/(\Delta \log \log n), n^{-4/5}\})$.*

Cet exemple montre en particulier l'intérêt des réductions pour concevoir des résultats d'approximation. Toute amélioration de l'approximation de STABLE donnera immédiatement lieu à une amélioration de ce résultat.

6.5.5 Du stable à la clique

Cet exemple permet d'illustrer, la problématique des paramètres et des rapports dépendant de l'instance. Les problèmes STABLE et CLIQUE maximum sont connus pour être équivalents, une clique de G étant un stable sur le graphe complémentaire¹⁴ $\bar{G}$, et réciproquement. Chacun sait (paragraphe 2.2.2) qu'en passant du stable à une couverture de sommets (complémentaire d'un stable) la valeur objective est composée par une fonction affine, ce qui rend ces problèmes non équivalents pour le rapport γ (mais équivalents pour le rapport δ). Par contre, lors du passage de CLIQUE à STABLE, ni la valeur objective ni la valeur optimale ne changent et le rapport est exactement le même pour les deux problèmes. En repensant soudain à la réduction du paragraphe 6.5.3 le lecteur attentif se souvient qu'il faut prendre garde à la taille des instances auxquelles on applique un algorithme, mais dans le cas présent G et $\bar{G}$ ont exactement la même taille, d'où l'égalité du rapport, ... À cela vous répondrez « Je ne sais pas, tu as peut être raison ou peut être tort : tout dépend des paramètres intervenant dans le rapport d'approximation » ! En effet vous avez raison : tant que les résultats d'approximation sont exprimés en fonction de la taille n (le seul cas pris en compte dans les formalismes usuels), cette réduction préserve le rapport d'approximation de sorte que les problèmes STABLE et CLIQUE sont équivalents du point de vue des rapports fonction de n . Par contre, lors du passage de G à $\bar{G}$, le degré n'est pas préservé de sorte que cette réduction transforme un algorithme à rapport $\rho(\Delta)$ pour STABLE en un algorithme à rapport $\rho(\bar{\Delta})$ où $\bar{\Delta}$ désigne le degré du graphe complémentaire, à savoir $n - \delta_{\min}$ où $\delta_{\min}$ désigne le degré minimum de G . Votre démonstration fut brillante, vous avez réussi à faire douter votre interlocuteur : ces problèmes sont-ils équivalents ? Voilà un élément de réponse.

Théorème 6.

- 1° *Il existe une réduction l-P de CLIQUE à STABLE qui transforme un rapport $\rho(n, \Delta)$ pour STABLE en $\rho(\Delta + 1, \Delta)$ pour CLIQUE.*
- 2° *Il existe une réduction de CLIQUE à lui-même qui transforme un rapport $\rho(n, \Delta)$ en $\rho(\Delta + 1, \Delta)$.*

Preuve. Supposons un algorithme **A** pour STABLE garantissant le rapport $\rho(n, \Delta)$. Soit alors $G = (V, E)$ une instance de CLIQUE. Pour chaque $v \in V$, soit G_v le sous-graphe de G induit par $\{v\} \cup \Gamma(v)$ ($\Gamma(v)$ est le voisinage de v). Le graphe G_v est d'ordre $\Delta + 1$ et le degré de $\bar{G}_v$ est au plus Δ . Donc en appliquant **A** à $\bar{G}_v$, on détermine une clique K'_v de G_v garantissant $|K'_v| \geq \rho(\Delta + 1, \Delta)|K_v^*|$ où K_v^* désigne une clique optimale de G_v . En notant K^* une clique optimale de G , on a : $\forall v \in K^*, K_v^* = K^*$. Par conséquent, il suffit de prendre, parmi toutes les solutions K'_v , $v \in V$, une clique de taille maximum pour garantir le rapport $\rho(\Delta + 1, \Delta)$ par rapport à K^* , ce qui conclut la preuve du point 1.

Le point 2 se montre de la même manière. On remarque d'ailleurs que le résultat reste valable pour le cas pondéré. ■

Le point 2 signifie que pour CLIQUE, les paramètres n et Δ sont équivalents. Le point 1, quant à lui, correspond à une amélioration du rapport lorsqu'on passe de STABLE à CLIQUE.

14. Les arêtes de G sont des non-arêtes de $\bar{G}$ et réciproquement.

Répondre à la question si STABLE et CLIQUE sont équivalents (au moins pour le paramètre Δ) revient à chercher si les paramètres n et Δ sont équivalents pour STABLE. Le résultat suivant apporte une réponse négative.

Proposition 16. *Si $P \neq NP$, alors pour toute fonction $g : \mathbb{R}^+ \rightarrow \mathbb{R}^+$, aucune réduction polynomiale de STABLE à lui-même ne peut transformer un rapport $\rho(n)$ en $\rho(g(\Delta))$.*

Preuve. Supposons qu'une telle réduction existe et considérons l'algorithme recherchant un meilleur stable parmi ceux de taille inférieure ou égale à $g(3)$. Il garantit, pour tout graphe, le rapport $\rho(n) = 1$ si $n \leq g(3)$ et $g(3)/n$ sinon. Pourtant, aucun algorithme polynomial ne peut, pour cette fonction ρ , garantir $\rho(g(\Delta))$: un tel algorithme résoudrait de manière exacte STABLE sur les graphes de degré 3 alors que ce problème est NP-complet ([28]). ■

6.5.6 Réductions entre les cadres classique et différentiel

La relation algébrique entre les valeurs des rapports différentiels et classiques permet, dans certains cas, de transformer immédiatement un résultat pour l'un des rapports en un résultat pour l'autre. Le cas le plus immédiat est celui des problèmes (de maximisation) pour lesquels la pire valeur est nulle (par exemple STABLE) puisqu'alors les deux rapports coïncident. Un exemple un peu plus élaboré concerne des problèmes de minimisation pour lesquels il existe une constante $\epsilon \in]0, 1[$ telle que $\forall I, \omega(I) \geq \beta(I)/(1 - \epsilon)$. On montre alors aisément : si $\gamma(I) \geq \sigma$, alors $\delta(I) \geq (\sigma + \epsilon - 1)/\sigma\epsilon$. Cette situation correspond à une réduction du problème à lui-même transformant un rapport classique σ en rapport différentiel $(\sigma + \epsilon - 1)/\sigma\epsilon$. Dans ce cas, les fonctions f et h , dans la définition 7, sont des fonctions identité. Des règles similaires peuvent être établies pour des problèmes de maximisation et pour le passage de δ à γ . Un exemple simple de ce passage concerne tous les problèmes de maximisation à valeurs de solutions strictement positives. En effet, considérons un tel problème admettant un algorithme à rapport différentiel δ . Un calcul simple permet dans ce cas d'établir que si $\delta(I) \geq \delta$, alors $\gamma(I) \geq \delta$. Cependant, la diversité des comportements qu'on observe entre les approximations classique et différentielle d'un même problème montre que le cas général n'est pas aussi simple.

Pour BINPACKING, par exemple, un tel lien ne peut exister. En effet, considérons une instance contenant $2n$ nombres tous égaux à $1/2$. La valeur optimale est n (deux nombres par *bin*). Mais alors, placer un nombre par *bin* définit une solution réalisable de valeur $2n$ donc garantissant, pour cette instance, le rapport $1/2$. Or, du point de vue différentiel, la garantie offerte par cette solution est nulle.

Dans [18], nous établissons une réduction de BINPACKING à lui-même permettant de déduire une approximation différentielle à partir d'un algorithme garantissant un rapport classique constant. Cependant, il s'agit d'une réduction l-P. Le résultat précis est le suivant.

Théorème 7. ([18]) *Il existe une réduction l-P de BINPACKING à lui-même transformant un rapport classique constant ρ en un rapport différentiel $(2 - (1/\rho))$.*

Ce résultat combiné au rapport classique $1 - 1/\beta(I) - 1/k$ est le principal ingrédient permettant d'établir un schéma d'approximation différentiel pour BINPACKING ([18]).

7 Difficulté des instances d'un problème

Nous concluons ce travail en abordant la question de la difficulté intrinsèque des instances d'un problème. Les concepts que nous avons discutés permettent de concevoir une structure de l'ensemble des problèmes. Pourtant, comprendre, pour un problème donné, la structure de ses instances a déjà un grand intérêt :

- des points de vue épistémologique et mathématique d'abord, tenter de caractériser la difficulté des instances d'un problème est une voie vers une meilleure perception de schémas qui rendent un problème accessible ou non à une résolution algorithmique ;

- d'un point de vue opérationnel maintenant, mettre en évidence, parmi les instances d'un problème, les structures qui donnent lieu à une difficulté de résolution ou qui mettent en défaut un algorithme précis est une étape essentielle pour analyser et améliorer ce dernier.

Parmi les notions que nous avons discutées dans ce document, celles d'*ordre de difficulté* et de *réduction* permettent d'aborder cette question. La première correspond à une tentative de description absolue de la structure des instances et d'une mesure quantitative de leur difficulté. Elle met plutôt l'accent sur la géométrie de l'ensemble des instances. L'exploitation de réductions est une manière duale d'aborder cette problématique. En associant à un ensemble d'instances d'un problème Π un sous-problème, cette approche consiste à comparer, à l'aide de réductions, différents sous-problèmes de Π ou encore de comparer Π à l'un de ses sous-problèmes. Il s'agit donc plutôt d'une « description relative » de l'ensemble des instances fondée sur la comparaison de leur difficulté.

Dans cette partie, nous étudions ces deux approches sur la base de plusieurs exemples. Ils mettent en évidence les liens et la complémentarité de l'approche absolue et de l'approche relative. La première a l'intérêt de fournir une mesure de difficulté et permet, par l'analyse de la structure des instances d'un problème, de déduire des propriétés sur son approximation. La seconde par contre est souvent plus souple et a l'intérêt d'être directement formulée de manière algorithmique.

Il est facile de se rendre compte que la mise en évidence d'une classe d'instances $\mathcal{I}' \subset \mathcal{I}$ se comportant mieux du point de vue de l'approximation que le problème dans son ensemble donne immédiatement accès à une réduction du problème général au sous problème correspondant aux instances $\mathcal{I} \setminus \mathcal{I}'$. C'est notamment le cas lorsque $\mathcal{I}'$ est l'ensemble des instances dont l'ordre de difficulté ne dépasse pas une valeur fixée k .

Par contre, il paraît nettement moins aisé de déduire, à partir d'une réduction d'un problème à un sous problème, une mesure de difficulté ; nous l'illustrons dans le paragraphe 7.4. Néanmoins, les exemples du paragraphe 7.3 illustrent que ce passage est parfois possible et intéressant.

7.1 Deux notions de simplicité

Lorsqu'elle s'applique, la notion d'ordre de difficulté a l'avantage de fournir une mesure de difficulté par rapport à l'approximation. Une telle possibilité est assez séduisante car le paramètre « ordre de difficulté » joue alors le rôle de variable explicative ou indicative permettant d'orienter la résolution. Cependant, comme nous le verrons dans ce paragraphe, l'ordre de difficulté n'est pas toujours calculable en temps polynomial par rapport à la taille de l'instance. Cette approche consistant à déterminer à partir de propriétés structurelles des instances d'un problème ses propriétés d'approximation se situe pleinement dans la continuité de travaux menés dès le début des années 80 et qui ont essentiellement conduit à la caractérisation combinatoire des classes **PTAS** et **FPTAS**.

Nous avons déjà vu, dans le paragraphe 4.2.3, que des notions très classiques de difficulté telles que la **NP**-complétude et la **NP**-complétude au sens fort donnent lieu à des exemples immédiats d'ordre de difficulté. Deux autres notions ont joué un rôle important ; il s'agit des deux notions de simplicité introduites respectivement dans [5] et dans [55]. Toutes deux s'expriment en termes d'ordre de difficulté.

La simplicité au sens de [55] correspond aux problèmes **NP**-difficiles pour lesquels la restriction aux instances I dont la valeur optimale $\beta(I)$ est majorée par une constante constitue un problème polynomial. Il s'agit donc exactement des problèmes pour lesquels la valeur optimale β est un ordre de difficulté par rapport à la résolution exacte. Dans le même esprit, en notant $\omega(I)$ la pire valeur (au sens de l'objectif) de l'instance I et $\beta(I)$ sa valeur optimale, un problème **NP**-difficile est *simple* au sens de [4] si $d(I) = |\beta(I) - \omega(I)|$ est un ordre de difficulté pour la résolution exacte.

Les problèmes STABLE ou COLORATION sont simples au sens de [4]. Pour s'en convaincre, il suffit de voir que les instances de ces problèmes pour lesquelles $|\beta(I) - \omega(I)|$ est majorée par une constante k peuvent être traitées par recherche exhaustive en temps $O(n^k)$. S reste simple au sens de [55], par contre C ne l'est pas puisque la k -colorabilité, pour $k \geq 3$ est un problème **NP**-complet ([28]). Dans le paragraphe 7.2 nous verrons d'autres exemples de problèmes non-simples.

La proposition suivante constitue le point de départ de la caractérisation de la classe **PTAS** et motive ces deux définitions.

Proposition 17.

- 1° Soit Π un problème d'optimisation **NP**-difficile dont l'objectif est à valeurs entières et admettant un schéma d'approximation polynomial, alors β est un ordre de difficulté pour la résolution exacte ([55]).
- 2° Soit Π un problème d'optimisation **NP**-difficile dont l'objectif est à valeurs entières et admettant un schéma d'approximation polynomial différentiel, alors $|\beta(I) - \omega(I)|$ est un ordre de difficulté pour la résolution exacte ([49]).

Preuve. Considérons le point 1 pour un problème de maximisation ; les autres cas sont similaires. Soit une constante k , on déduit du schéma d'approximation polynomial un algorithme polynomial garantissant le rapport $1 - 1/(k + 2)$. Soit I une instance telle que $\beta(I) \leq k$, soit $\lambda(I)$ la valeur de la solution approchée, elle vérifie $\lambda(I) \leq \beta(I) \leq \lambda(I)(k + 2)/(k + 1) < \lambda(I) + 1$, d'où l'égalité $\lambda(I) = \beta(I)$ puisque ces quantités sont entières. ■

Cette proposition très simple ne constitue évidemment pas une condition nécessaire et suffisante d'appartenance à la classe **PTAS** ; en particulier, STABLE est simple au sens de [55] mais n'admet pas de schéma d'approximation polynomial. Cependant, associée à des conditions de bornes, elle constitue une condition nécessaire et suffisante d'existence de schéma d'approximation ([55]) et de schéma d'approximation différentiel ([49]) pour des problèmes à valeurs entières. Ces caractérisations ne peuvent être reprises ici par souci de lisibilité du document mais sont dans le même esprit que la proposition 17.

7.2 Problèmes radiaux

Actuellement aucune caractérisation combinatoire n'est connue pour des classes d'approximation plus larges que **PTAS**, notamment pour **APX** ou des classes d'approximation non-constante. Toutefois, un point de vue intéressant pour obtenir des conditions suffisantes est de concevoir des classes de problèmes (ou problèmes générique) fondées sur des propriétés combinatoires pour lesquelles on établit des résultats d'approximation.

7.2.1 Problèmes héréditaires

Un exemple significatif est celui du problème générique HG consistant à *maximiser un ensemble satisfaisant une propriété héréditaire vérifiable en temps polynomial*. Soit E un ensemble fini ; nous rappelons qu'une propriété $\pi : 2^E \rightarrow \{0, 1\}$ sur l'ensemble 2^E des parties de E est *héréditaire* si $A \subseteq B$ implique $\pi(A) \geq \pi(B)$. En d'autres termes π est transmise à toute partie d'un ensemble dès lors qu'elle est satisfaite pour l'ensemble. De telles propriétés sont très naturelles notamment dans des circonstances satisfaisant la loi du « qui peut le plus peut le moins ». C'est en particulier le cas pour des problèmes où le caractère réalisable d'un ensemble de décisions est conditionné par la saturation d'une ressource. Ainsi, par exemple, la propriété, pour un ensemble de sommets d'un graphe, d'être stable est héréditaire : c'est le cas où les seules incompatibilités sont des incompatibilités par paires. De même, les propriétés d'être une clique (sous-graphe complet), d'être de degré inférieur à une valeur fixée, d'être k -colorable, ..., sont héréditaires, mais c'est aussi le cas de bien d'autres propriétés plus ou moins complexes. Par conséquent, cette

classe de problèmes est très large et contient entre autres les problèmes STABLE et CLIQUE. Un résultat d'approximation portant sur cette classe apparaît alors comme une condition suffisante d'appartenance à un certain niveau d'approximation. Ainsi, par exemple, le résultat suivant est montré dans [31].

Théorème 8. *HG est approximable au niveau $O(\log n/n)$ où n est le cardinal de E .*

7.2.2 Une généralisation

La notion de problème radial est une généralisation de la propriété d'hérédité qui permet d'englober beaucoup d'autres problèmes, et notamment des problèmes de minimisation.

Étant donné un problème **NPO**, le *support* d'une instance I , noté $\sigma(I)$, est défini comme le nombre de valeurs réalisables de I . Cette notion est naturelle dans l'optique de capter la difficulté d'une instance et, en particulier, si on cherche à étudier la capacité qu'a une instance à être bien résolue par un algorithme glouton ou encore par programmation dynamique.

Le principe d'un algorithme glouton est de construire une solution par choix élémentaires successifs (reposant sur un critère de préférence) sans retour en arrière. Une telle démarche peut être représentée comme une succession d'améliorations à partir d'une solution triviale (par exemple l'ensemble vide pour un problème de maximisation). Les instances de support élevé s'interprètent comme les cas les plus défavorables puisque la construction d'une bonne solution à partir de la solution de base nécessite alors (dans le pire des cas) de nombreuses étapes gloutonnes. Cet argument peut être précisé grâce au formalisme que nous venons d'introduire. Pour cela, nous nous restreignons à une classe - très large - de problèmes (appelés *radiaux*) dont l'intérêt est d'offrir un cadre générique pour analyser des démarches de type gloutonnes. Elle généralise en particulier les problèmes *héréditaires* pour lesquels toute partie d'une solution réalisable est réalisable.

Considérons un problème d'optimisation Π (définition 1) et rappelons que chaque instance de Π s'exprime par :

$$\begin{cases} \text{opt} & v(x) \\ & x \in \mathcal{C} \\ & x_i \in \{0, 1\}, \end{cases}$$

où opt vaut respectivement $\max$ ou $\min$ selon que Π est un problème de maximisation ou de minimisation. On définit $\bar{\theta} = <$ (resp., $>$) si $\text{opt} = \max$ (resp., $\min$). Nous proposons alors la définition suivante.

Définition 8. Un problème Π est *radial* s'il existe trois algorithmes polynomiaux (en n) ξ , ψ et φ tels que pour toute instance I de Π de taille n :

- 1° ξ construit une solution réalisable $x^{(0)}$,
- 2° pour toute solution réalisable x de I strictement meilleure que $x^{(0)}$, l'algorithme φ construit une solution réalisable $\varphi(x)$ qui vérifie $v(\varphi(x))\bar{\theta}v(x)$,
- 3° pour tout vecteur réalisable x de I strictement meilleure que $v(x^{(0)})$, il existe un entier k tel que $\varphi^k(x) = x^{(0)}$ (φ^k désignant l'itéré k -fois de φ),
- 4° pour $x^{(0)}$ ainsi que pour toute solution réalisable x de I de valeur objective strictement meilleure que $v(x^{(0)})$, $\psi(x) = \varphi^{-1}(\{x\}) = \{y, \varphi(y) = x\}$. ■

Géométriquement, $x^{(0)}$ et les solutions qui lui sont strictement meilleures peuvent être distribuées sur un arbre planaire de racine $x^{(0)}$ de sorte que la valeur objective augmente strictement sur un chemin de la racine vers les feuilles. Dans cet arbre $\varphi(x)$ est le père de x et $\psi(x)$ est l'ensemble de ses fils. Dans ce contexte chaque itération d'un algorithme glouton correspond à sélectionner un fils ($\psi(x)$ est l'ensemble des solutions pouvant être construites à partir de x). Il s'agit de problèmes résolubles par une machine de TURING polynomiale non déterministe de type glouton.

Pour de tels problèmes une résolution gloutonne à partir de $x^{(0)}$ n'exclut a priori aucune solution réalisable meilleure que $x^{(0)}$.

Les problèmes héréditaires sont l'exemple le plus classique de cette situation : la solution de départ $x^{(0)}$ est la solution vide correspondant au vecteur $\vec{0}$ et l'opération élémentaire représentée par l'algorithme φ est le retrait d'un élément. Un algorithme glouton consiste alors à construire une solution par sélections successives. De très nombreux autres problèmes non héréditaires entrent dans le cadre de la définition 8. Citons notamment H-TRANSVERSAL ou G-TRANSVERSAL, BINPACKING ou encore COLORATION. Par contre, le problème de la coloration minimum d'un graphe 4-colorable¹⁵ ou encore le problème BINPACKING_C défini au paragraphe 4.2.1 ne sont pas radiaux (la proposition 19 permet de le montrer).

Lorsque $\sigma(I)$ est borné par une constante, l'arbre associé à une instance est de profondeur bornée et alors, sous les hypothèses de radialité, le parcours complet de cet arbre devient polynomial. On en déduit le résultat suivant.

Proposition 18. *$\sigma(I)$ est un ordre de difficulté pour la résolution exacte pour les problèmes radiaux NP-complets.*

7.2.3 Liens avec les notions de simplicité

Pour un problème d'optimisation, dès qu'une distance minimum ϵ sépare deux valeurs réalisables (en particulier pour les problèmes à valeurs entières), on a $\sigma(I) \leq (1/\epsilon)|\beta(I) - \omega(I)|$. On en déduit le résultat suivant.

Proposition 19. *Les problèmes radiaux NP-complets pour lesquels une distance minimum ϵ sépare deux valeurs réalisables sont simples au sens de [4].*

À l'inverse, le problème de la satisfiabilité pondérée maximum (*max-weighted-sat* ([4])) ne l'est pas. Il en est de même pour la coloration minimum d'un graphe 4-colorable ou encore pour le problème BINPACKING_C. Ainsi, la proposition 19 constitue un test simple de non-radialité.

Les problèmes pondérés fournissent facilement des exemples de problèmes radiaux non-simples. En effet, en général les pondérations n'affectent pas le caractère radial d'un problème qui représente plutôt la structure d'une instance ; la simplicité, par contre, est tributaire des poids. Par exemple, WSTABLE (les poids sont rationnels positifs) est radial. Par contre, en attribuant à chaque sommet le poids $1/n$ (n est le nombre de sommets), on borne artificiellement l'objectif par 1 sans modifier la difficulté du problème ; WS n'est donc pas un problème simple.

Par contre, il est plus délicat de trouver des problèmes simples mais non radiaux. Un second critère de non-radialité est la possibilité, pour une solution réalisable, d'avoir un nombre non polynomial de fils dans l'arbre associé. Ainsi, un problème pour lequel borner l'objectif par des constantes rend la résolution polynomiale mais laisse un nombre non-polynomial de solutions réalisables est simple mais non radial. C'est d'ailleurs ainsi qu'on peut facilement concevoir des versions non-radiales mais polynomiales du problème de couplage maximum (par exemple en imposant une borne inférieure aux solutions réalisables).

Voici un autre exemple fondé sur STABLE. Considérons le problème $\tilde{S}$ dont une instance est la donnée : d'un graphe G et d'un ensemble stable S_0 tel qu'il existe un stable maximum S^* de G contenant S_0 . L'objectif est de déterminer un stable maximum de taille au moins égale à $|S_0|$.

Pour ce problème, la pire valeur d'une instance est $|S_0|$ vue comme une valeur réalisable offerte avec l'instance. La valeur optimale est le nombre de stabilité de G . Le problème $\tilde{S}$ sort du cadre habituel de NPO car ses instances ne peuvent pas être reconnues en temps polynomial.

15. Une instance de ce problème étant la donnée d'un graphe G et d'une 4-coloration de G , les colorations réalisables ayant au moins 4 couleurs.

En effet, pour un graphe $G = (V, E)$ et un entier $k \in \{1, \dots, n\}$ on définit le graphe $G_k = (V^k, E^k)$ où :

$$\begin{cases} V^k &= V \cup \{1, \dots, k\} \\ E^k &= E \cup \{vi \in V \times \{1, \dots, k\}\} \end{cases}$$

En d'autres termes, G_k est obtenu en ajoutant à G k sommets constituant un stable et toutes les arêtes entre ces sommets nouveaux et les sommets de G ; G_k peut être construit en temps polynomial et en posant $S_0 = \{1, \dots, k\}$ (ensemble des nouveaux sommets), il est clair que (G_k, S_0) est une instance de $\tilde{S}$ si et seulement si k est inférieur ou égal au nombre de stabilité de G . Un algorithme reconnaissant les instances de $\tilde{S}$ pourrait ainsi être exploité pour résoudre la version décision de STABLE¹⁶.

Néanmoins, certains formalismes au delà de **NPO** permettent d'envisager ce type de problème. Remarquons par ailleurs que $\tilde{S}$ est au moins aussi difficile que STABLE. On peut alors facilement établir le résultat suivant.

Proposition 20. *$\tilde{S}$ est simple au sens de [4] mais non radial.*

Preuve. Pour montrer que $\tilde{S}$ est simple, il suffit de se rendre compte que, si $|S^*| - |S_0|$ est majoré par une constante C , alors une recherche exhaustive à partir de S_0 de tous les ensembles de taille au plus $|S_0| + C$ sommets contenant S_0 se fait en temps polynomial et permet de déterminer toutes les solutions optimales contenant S_0 .

Remarquons cependant que ces instances, bien que résolues en temps polynomial, ne sont pas reconnues en temps polynomial. Pour imposer le caractère polynomial de la reconnaissance de ces instances faciles, il faudrait en outre inclure dans chaque instance la valeur d'une solution optimale. Pour ce nouveau problème noté $\tilde{S}'$, on pourrait expliciter en quel sens il n'est pas résoluble en temps polynomial. Il resterait simple car pas plus difficile que $\tilde{S}$.

Pour montrer, maintenant, le caractère non-radial de $\tilde{S}$, il suffit de montrer que même pour les instances telles que $|S^*| - |S_0|$ est majoré par une constante C , on ne peut pas déterminer toutes les solutions réalisables en temps polynomial. En effet, si un tel algorithme **A** existait, il suffirait, d'appliquer la démarche suivante pour déterminer, en temps polynomial, un stable maximum d'un graphe G .

BEGIN

$S \leftarrow \emptyset$;

$k \leftarrow n + 1$;

REPEAT

$k \leftarrow k - 1$;

construire G_k ;

$S_0 \leftarrow \{1, \dots, k - 1\}$;

appliquer **A** à (G_k, S_0) ;

IF G_k admet un stable optimal $V' \subseteq V$ THEN $S \leftarrow V'$ FI

UNTIL $S \neq \text{emptyset}$;

END.

Remarquons que toutes les instances auxquelles **A** est appliqué sont des instances de $\tilde{S}$ pour lesquelles $|S^*| - |S_0| \leq 1$. L'algorithme finit dès que k vaut le nombre de stabilité de G et dans ce cas, S est un stable optimal. Cet argument pourrait immédiatement être appliqué à $\tilde{S}'$. ■

7.3 Structuration par élargissement d'une classe

La notion d'ordre de difficulté permet d'ordonner les instances de problèmes donnés à partir de leurs propriétés d'approximation. Dans la précédente section, nous avons envisagé une situation

¹⁶. Elle correspond à la question suivante : étant donnés G et k , existe-t-il un stable de G de valeur au moins k ?

un peu différente en introduisant une classe de problèmes définie par une structure spécifique de leurs instances, cette structure permettant alors a posteriori d'obtenir des informations sur les possibilités de résolution. Une démarche intermédiaire consiste, pour un problème donné, à définir une classification des instances à partir de propriétés combinatoires non directement liées à une valeur de paramètre et d'en déduire des résultats d'approximation sur chaque classe. La structure ainsi obtenue peut, dans certains cas, apparaître comme une réelle hiérarchie des instances par rapport à leurs possibilités de résolution ; les caractéristiques combinatoires qui ont permis de la définir sont alors, en un certain sens, explicatives de la difficulté. Dans le cas le plus favorable ces propriétés combinatoires peuvent être liées à une valeur de paramètre et correspondre exactement à un ordre de difficulté. Dans d'autres cas, seule une partie de la hiérarchie est le siège de propriétés combinatoires intéressantes ou encore, ces propriétés ne portent que sur un algorithme spécifique ou une classe d'algorithmes.

Cette section s'inscrit dans le cadre de cette situation intermédiaire. Plus précisément, nous nous intéressons à des problèmes pour lesquels une classe particulière d'instances $\mathcal{I}_0$ est connue pour son bon comportement par rapport à l'approximation. L'objectif est alors de structurer les instances par rapport à cette classe en vue d'interpréter ce « bon comportement » comme un cas particulier très favorable d'un phénomène dépassant la classe $\mathcal{I}_0$.

7.3.1 Stabilité d'un algorithme par rapport à une fonction d'éloignement

La notion de *stabilité en approximation* a été introduite dans [9]. Pour un problème Π d'ensemble d'instances $\mathcal{I}$ et un sous-problème Π' d'ensemble d'instances $\mathcal{I}' \subset \mathcal{I}$, on appelle fonction d'éloignement par rapport à $\mathcal{I}'$ une fonction h calculable en temps polynomial associant à toute instance I de $\mathcal{I}$ un nombre positif qui est nul si $I \in \mathcal{I}'$. On définit alors une notion de *halo autour de $\mathcal{I}'$ de rayon r par rapport à h* , noté $B_{r,h}(\mathcal{I}')$, l'ensemble des instances I telles que $h(I) \leq r$.

Plaçons-nous dans la situation où un algorithme A initialement adapté à la classe d'instances $\mathcal{I}'$ et garantissant, sur cette classe, un certain niveau d'approximation puisse en fait s'appliquer à toute instance de $\mathcal{I}$ en fournissant, en temps polynomial, une solution réalisable (a priori sans garantie). La notion de stabilité ([9]) évoque alors l'idée de « continuité de la garantie de performance au voisinage de $\mathcal{I}'$ ». En effet, il s'agit d'étudier comment se détériore la qualité de A lorsqu'on l'applique à des instances hors de $\mathcal{I}'$. Dans le cas où A garantit sur $\mathcal{I}'$ un rapport constant, la stabilité définie dans [9] correspond au cas où A garantit un rapport constant sur tout halo de rayon r . Si le problème global Π n'est pas dans **APX**, ceci implique que la fonction d'éloignement h est un ordre de difficulté par rapport à l'approximation à rapport constant.

Dans [9] cette notion est exploitée sur l'exemple de TSP. Il est connu pour être approximable à rapport constant dans le cas où l'inégalité triangulaire est satisfaite (on note Δ -TSP le sous-problème correspondant), alors qu'un tel résultat ne peut être garanti pour toute instance. La stabilité de plusieurs algorithmes classiques est étudiée par rapport à deux fonctions d'éloignement par rapport à la classe $\mathcal{I}'$ des instances vérifiant l'inégalité triangulaire. La conception d'un algorithme stable permet alors notamment d'obtenir un algorithme à rapport constant pour la classe Δ_β -TSP (pour une constante β) des graphes tels que, pour tout triangle, le coût d'un coté ne dépasse pas β fois la somme des coûts des deux autres cotés.

Une manière de généraliser cette notion est de parler de stabilité lorsque Π se réduit à Π' avec une expansion de dépendant uniquement de la valeur de la fonction d'éloignement. Dans le cas, moins favorable, où l'expansion dépend d'autres paramètres que la fonction d'éloignement on parle ([9]) de quasi-stabilité.

7.3.2 Autour de la propriété de KÖNIG-EGERVARY

Ce paragraphe est un autre exemple d'élargissement d'une classe et de résultats algorithmiques s'y rapportant. Il ne s'agit pas à proprement parler d'un cas de stabilité car la complexité de l'algorithme étendu dépend de l'éloignement par rapport à la classe initiale. Dans cet exemple l'extension est rendue possible par une interprétation combinatoire de la fonction d'éloignement.

La propriété de KÖNIG-EGERVARY pour un graphe correspond à l'égalité entre le nombre de stabilité (noté $\alpha(G)$) et la cardinalité d'une couverture minimum d'arêtes (à savoir $n - m$ où n est le nombre de sommets et m le cardinal d'un couplage maximum). Il est bien connu que la restriction de STABLE à cette classe est polynomiale. La différence $n - m - \alpha(G)$ peut être vue comme un *saut de dualité discret*. Exprimons en effet STABLE par le programme linéaire suivant :

$$\text{STABLE} = \begin{cases} \max & \vec{1}_{|V|} \cdot \vec{x} \\ & A \cdot \vec{x} \leq \vec{1}_{|E|} \\ & \vec{x} \in \{0, 1\}^{|V|} \end{cases}$$

où $\vec{1}_k$ est le vecteur colonne de $\mathbb{Q}^k$ dont toutes les coordonnées valent 1 et A est la matrice arêtes-sommets de G .

Exprimons également le problème de couverture minimum d'arêtes par le programme :

$$\text{EC} = \begin{cases} \min & \vec{1}_{|E|} \cdot \vec{x} \\ & A^T \cdot \vec{x} \geq \vec{1}_{|V|} \\ & \vec{x} \in \{0, 1\}^{|E|} \end{cases}$$

Sa valeur optimale vaut $v^*(\text{EC}) = n - m$.

On remarque que les relaxés de ces problèmes (pour lesquels les relations $x_i \in \{0, 1\}$ sont remplacées par $x_i \geq 0$) sont des programmes linéaires duaux.

La valeur $D(G) = n - m - \alpha(G)$ du saut de dualité discret peut jouer le rôle d'une fonction d'éloignement par rapport à la classe des graphes KÖNIG-EGERVARY, notée KE. Se pose alors la question de la difficulté de STABLE en fonction de cet éloignement.

Pour y répondre on commence par établir la proposition suivante qui est une interprétation combinatoire de la fonction d'éloignement.

Soit $G = (V, E)$ un graphe, M un couplage maximum de cardinal m , X l'ensemble des sommets exposés par rapport à M , S^* un stable maximum et $C^* = V \setminus S^*$ la couverture de sommets optimale associée à S^* . On note alors F (de cardinal f) l'ensemble des arêtes de M dont les deux extrémités sont dans C^* et X' (de cardinal g) l'ensemble $X \cap C^*$. Les ensembles F et X' dépendent des ensembles M et S^* fixés, par contre, la quantité $f + g$ est indépendante du couplage maximum et du stable maximum choisis : on a $f + g = D(G) = n - m - \alpha(G)$.

Les graphes KÖNIG-EGERVARY correspondent au cas $D(G) = 0$, ce qui signifie ($f + g = 0$) que tous les sommets exposés sont dans S^* et que chaque arête de M a une extrémité dans S^* .

Proposition 21. *$D(G)$ est un ordre de difficulté pour la résolution exacte.*

Preuve. Ce résultat provient immédiatement de l'interprétation combinatoire de la valeur $D(G)$: $f + g \leq k$ pour k constant implique $f \leq k$ et $g \leq k$, par conséquent on peut énumérer (en temps constant) tous les couples d'ensembles (Φ, Γ) tels que $\Phi \subset M, \Gamma \subset X, |\Phi| + |\Gamma| \leq k$. Pour l'un d'eux au moins (celui correspondant à $\Phi = F$ et $\Gamma = X'$) le graphe induit par $V \setminus (\Phi[V] \cup \Gamma)$ où $\Phi[V]$ désigne l'ensemble des extrémités des arêtes de Φ , est KÖNIG-EGERVARY. Il suffit donc d'appliquer à chacun de ces sous-graphes induits un algorithme de reconnaissance de la propriété de KÖNIG-EGERVARY et de résolution de STABLE lorsqu'elle est vérifiée. ■

Remarquons que $D(G) \geq n/2 - \alpha(G)$. Par ailleurs, nous montrons dans la proposition 25 de la section 7.4.3 qu'on peut toujours, pour la résolution de STABLE, se ramener au cas où $\alpha(G) \leq n/2$.

Dans ce cas, une question naturelle est de savoir si la quantité $|n/2 - \alpha(G)| \leq D(G)$ est un ordre de difficulté pour la résolution exacte. La réponse est négative car la restriction de STABLE au cas $\alpha(G) = n/2$ est **NP**-complète. En effet, en supposant $\alpha(G) \leq n/2$, il suffit d'ajouter k sommets isolés pour garantir au graphe résultant d'avoir un nombre de stabilité $k + \alpha(G)$ pour $n + k$ sommets. En appliquant ce processus pour tout $k \in \{0, \dots, n\}$ (donc en particulier pour $k = n - 2\alpha(G)$) il serait possible, à partir d'un algorithme polynomial garantissant une solution optimale pour les graphes dont le nombre de stabilité vaut la moitié des sommets, de résoudre le problème général. Dans le paragraphe 7.4.2, nous discutons les propriétés d'approximation d'une classe proche qui comprend les graphes dont le nombre de stabilité vaut au moins $n/2$ (ou plus généralement au moins n/k).

Si on autorise maintenant à $D(G)$ de prendre des valeurs non constantes, on parvient encore à maîtriser les possibilités de résolution efficace (au sens de l'approximation) en fonction de la valeur de $D(G)$.

Proposition 22. *Il existe un algorithme polynomial pour STABLE garantissant le rapport $1/(D(G) + 1)$.*

Preuve. Considérons l'algorithme suivant :

```

BEGIN
(1)  construire un couplage maximum M;
(2)   $m \leftarrow |M|$ ;
(3)   $x \leftarrow |X|$ ;
(4)   $S \leftarrow X$ ;
(5)  FOR  $k \in \{0, \dots, n-1\}$  DO
(6)    partitionner M en  $k+1$  parties  $M_1, \dots, M_{k+1}$  de taille  $\geq \lfloor m/(k+1) \rfloor$ ;
(7)     $\mathcal{M} \leftarrow \{M_i : i = 1 \dots k+1\}$ ;
(8)    Partitionner X en  $k+1$  parties  $X_1, \dots, X_{k+1}$  de taille  $\geq \lfloor x/(k+1) \rfloor$ ;
(9)     $\mathcal{X} \leftarrow \{X_i : i = 1 \dots k+1\}$ ;
(10)   FOR  $(A, B) \in \mathcal{M} \times \mathcal{X}$  DO
(11)     IF  $G' = G[X(A) \cup B] \in KE$  THEN
(12)        $S' \leftarrow$  stable maximum de  $G'$ ;
(13)       IF  $|A| < m/(k+1)$  AND  $|B| < x/(k+1)$  THEN
(14)         IF  $\exists u \in M \setminus A$  tel que  $G'' = G[X(A \cup \{u\}) \cup B] \in KE$  THEN
(15)            $S' \leftarrow$  stable maximum de  $G''$ ;
(16)         ELSE IF  $\exists v \in E \setminus B$  tel que  $G'' = G[X(A) \cup B \cup \{v\}] \in KE$  THEN
(17)            $S' \leftarrow$  stable maximum de  $G''$ ;
(18)         FI
(19)       FI
(20)     FI
(21)      $S \leftarrow \text{Argmax}\{|S|, |S'|\}$ ;
(22)   FI
(23) OD
(24) OD
END.
```

Remarquons d'abord que l'algorithme fonctionne en temps polynomial. En effet, la ligne (1) a la complexité du problème de couplage maximum qui est polynomial ([53]). La ligne (5) implique n boucles, pour chacune d'elles, la complexité maximum provient de la boucle (10) qui est exécutée au plus $(k+1)^2$ fois, avec, pour chaque exécution, $O(n^{2,5})$ étapes. Il en résulte une complexité globale de l'ordre de $O(n^{5,5})$.

Notons aussi que, pour deux entiers k et $l \neq 0$ on a

$$\frac{k}{l} - \left\lfloor \frac{k}{l} \right\rfloor \leq \frac{l-1}{l} \quad (9)$$

Soit une instance G correspondant aux valeurs n, m, e, D, f et h à laquelle on applique l'algorithme. Fixons un stable maximum de G correspondant aux ensembles F et X' de taille respective f et g . Il est clair que $D(G) \in \{0, \dots, n-1\}$; considérons alors l'exécution de la boucle **FOR** avec $k = D(G)$. On a $k+1 > f$ et $k+1 > g$, donc $\exists (M_i, X_j) \in \mathcal{M} \times \mathcal{X}$ tel que $M_i \cap F = \emptyset$ et $X_j \cap X' = \emptyset$. Mais alors, le graphe $G[X(M_i) \cup X_j]$ est dans la classe KE et est détecté à la ligne (11).

Pour ce couple (M_i, X_j) , si les lignes (14) à (17) sont exécutées et si aucun graphe KE n'est détecté ni à la ligne (14) ni à la ligne (16), alors $F = M \setminus M_i$ et $X' = X \setminus X_j$, ce qui signifie que $\alpha(G') = \alpha(G)$ et que S' est un stable maximum. Dans tous les autres cas, (la ligne (14) n'est pas exécutée ou l'une des lignes (15) ou (17) est exécutée), le stable S' vérifie (cf., expression (9))

$$|S'| \geq \frac{m}{D(G)+1} + \frac{|X|}{D(G)+1} - \frac{D(G)}{D(G)+1}. \quad (10)$$

Par ailleurs, $\alpha(G) = m + |X| - D(G)$. La solution construite étant de taille au moins égale à celle de S' , les expressions (10) et celle pour $\alpha(G)$ concluent que le rapport $1/(D(G)+1)$ est garanti. ■

Remarquons que dans le cas où $D(G) = 0$ (classe KE) on retrouve le rapport 1 correspondant à la résolution exacte. Ce résultat peut donc être interprété comme un algorithme stable par rapport à $D(G)$, vu comme fonction d'éloignement par rapport à la classe KE, à ceci près que $D(G)$ n'est pas calculable en temps polynomial. Ce résultat, à la différence de celui de la proposition 21, établit un résultat valable pour tout graphe qui met en évidence comment le rapport d'approximation se détériore en fonction de l'éloignement.

7.4 Réduction à des sous-problèmes : autres exemples concernant le stable

La section précédente nous a montré comment des classes d'instances bien résolues peuvent donner lieu à un paramètre intéressant mesurant l'éloignement par rapport à cette classe. Celui-ci apparaît alors comme un ordre de difficulté naturel. Cette démarche est applicable à de très nombreuses situations pour lesquelles on dispose d'une classe d'instances que l'on peut résoudre avec de meilleures garanties que le problème général. Toutefois, l'ordre de difficulté qui en résulte n'est pas toujours aussi pertinent. Nous donnons dans cette section des exemples, concernant **STABLE**, pour lesquels une formulation par réduction nous semble plus intéressante et plus exploitable qu'une formulation par ordre de difficulté.

7.4.1 Stable et rapports $O(n^{\epsilon-1})$

Considérons toujours **STABLE** pour lequel il est établi ([36]) que, sous une hypothèse hautement improbable, aucun algorithme polynomial ne peut garantir un rapport de la forme $O(n^{\epsilon-1})$, avec $\epsilon \in]0, 1[$. Or un tel rapport est trivialement garanti (par une solution de taille 1) pour la classe des graphes dont le nombre de stabilité n'excède pas $n^{1-\epsilon}$ avec $\epsilon \in]0, 1[$.

Par simples manipulations algébriques, il est alors facile de se rendre compte que, pour toute fonction f vérifiant :

- 1° $f :]0, 1[\rightarrow]0, +\infty[$;
- 2° f strictement croissante;
- 3° $\lim_{x \rightarrow 1} f(x) = +\infty$;

la fonction $F(G) = f(\log \alpha(G)/\log n)$ est un ordre de difficulté par rapport au niveau d'approximation correspondant aux rapports de la forme $O(1/n^{1-\epsilon})$, $\epsilon \in]0, 1[$.

Une démarche similaire est applicable à de nombreux résultats négatifs avec un seuil σ dès lors que la classe des instances vérifiant $\beta\theta f(\sigma)$ est facilement approximable avec le rapport σ (f est une fonction numérique et $\bar{\theta} = \leq$ pour les problèmes de maximisation et $\geq$ pour les problèmes de minimisation). Toutefois, cette transcription du résultat négatif comme ordre de difficulté n'apporte dans ce cas aucune information intéressante pour le problème, notamment à cause du fait que l'expression de l'ordre de difficulté ne correspond pas à un paramètre naturel du problème.

Par contre, la formulation en termes de réductions reste intéressante, notamment parce qu'elle met en évidence une classe d'instances difficiles sur lesquelles il faut se focaliser. Notons $\text{STABLE}_{\geq n^{1-\epsilon}}$ le sous problème de STABLE restreint aux graphes $G = (V, E)$ tels que $\alpha(G) \geq |V|^{1-\epsilon}$. La proposition suivante, qui se déduit immédiatement de la discussion ci-dessus, met en évidence l'intérêt de la classe $\text{STABLE}_{\geq n^{1-\epsilon}}$.

Proposition 23. *STABLE se réduit à $\text{STABLE}_{\geq n^{1-\epsilon}}$ par une réduction préservant le niveau d'approximation $O(n^{\epsilon-1})$.*

Notons que pour cet exemple, non seulement le degré de difficulté associé ne peut être déterminé en temps polynomial, mais surtout, la condition $\alpha(G) \leq n^{1-\epsilon}$ avec $\epsilon > 0$ (ou encore la condition $F(G) \leq k$, $k \in \mathbb{N}$) qui sépare les instances les mieux résolues des instances les plus difficiles n'est pas reconnaissable en temps polynomial. Pour les problèmes simples par exemple ; l'ordre de difficulté n'est pas non plus polynomialement calculable mais, au moins, la condition $\beta(I) \leq k$ est en général reconnaissable en temps polynomial.

Une autre conséquence du caractère non-polynomial du test $\alpha(G) \leq n^{1-\epsilon}$ est que le problème $\text{STABLE}_{\geq n^{1-\epsilon}}$ n'entre pas explicitement dans le cadre **NPO** (définition 1) car ses instances ne sont pas polynomialement reconnaissables contrairement aux instances d'un problème **NPO** (il s'agit d'une conséquence assez simple de la définition 1). Nous verrons dans le paragraphe suivant une autre classe de problèmes de stabilité qui sort du cadre de **NPO** pour des raisons similaires. En fait, de plus en plus de travaux exploitent des problèmes de ce type ; afin de les inclure dans un cadre cohérent, une classe plus large que **NPO** est définie dans [46].

7.4.2 La classe STABLE_k

Nous venons de formaliser, dans la proposition ci-dessus, le fait que les instances du problème STABLE les plus difficiles du point de vue de l'approximation sont les graphes ayant un nombre de stabilité supérieur à $n^{1-\epsilon}$ avec $\epsilon \in]0, 1[$. À l'inverse nous avons mis en évidence dans le paragraphe 7.3.2 une classe de graphes ayant un nombre de stabilité élevé pouvant être résolue en temps polynomial, à savoir les graphes dont le nombre de stabilité est proche (à une constante k près) de la valeur minimum d'une couverture d'arêtes. Remarquons que dans ce cas, le nombre de stabilité est au moins égal à $(n/2) - k$ puisqu'au moins $n/2$ arêtes sont nécessaires pour couvrir les n sommets d'un graphe.

Plus généralement une classe intéressante apparue dans divers travaux sur l'approximation (par exemple, [42, 2, 46, 10, 21]) est formée des graphes dont le nombre de stabilité est au moins égal à n/k où n est le nombre de sommets et k une constante. Nous notons STABLE_k cette classe. Remarquons que, comme pour la classe $\text{STABLE}_{\geq n^{1-\epsilon}}$ introduite dans le précédent paragraphe, les instances de STABLE_k ne peuvent être reconnues en temps polynomial.

Il ressort des articles [10, 2] que, pour toute constante k , il existe $\epsilon \in]0, 1[$ tel que STABLE_k est approximable au niveau $O(n^{\epsilon-1})$. En notant $S_{[n^{1-\epsilon}, n/k]}$ la classe des graphes dont le nombre de stabilité est compris entre $n^{1-\epsilon}$ et n/k pour ϵ et k fixés, on obtient le résultat suivant.

Proposition 24. *STABLE se réduit à $\text{STABLE}_{[n^{1-\epsilon}, n/k]}$ par une réduction préservant le niveau*

d'approximation $O(n^{\epsilon-1})$.

Cette réduction met en évidence une classe d'instances difficiles de STABLE.

7.4.3 La classe de NEMHAUSER et TROTTER

Considérons WSTABLE (stable maximum pondéré) et intéressons-nous de nouveau aux liens entre le problème exprimé comme programme linéaire en variables bivalentes et son relaxé (noté WSTABLE_r) qui est polynomial. Ces deux programmes linéaires sont définis de la façon suivante pour une instance composée d'un graphe $G = (V, E)$ et d'un système de poids $\vec{w} \in \mathbb{Q}^{|V|}$:

$$\text{WSTABLE} = \begin{cases} \max & \vec{w} \cdot \vec{x} \\ & A \cdot \vec{x} \leq \vec{1}_{|E|} \\ & \vec{x} \in \{0, 1\}^{|V|} \end{cases}$$

où $\vec{1}_{|E|}$ est le vecteur colonne de $\mathbb{Q}^{|E|}$ dont toutes les coordonnées valent 1.

$$\text{WSTABLE}_r = \begin{cases} \max & \vec{w} \cdot \vec{x} \\ & A \cdot \vec{x} \leq \vec{1}_{|E|} \\ & \vec{x} \in \mathbb{Q}^{|V|} \end{cases}$$

Parmi les résultats intéressants dans ce cadre, mentionnons celui de la solution semi-intégrale.

Proposition 25. ([51]) *Il existe un algorithme polynomial permettant, pour toute instance de WSTABLE, de partitionner V en trois ensembles V_1 , $V_{1/2}$ et V_0 de telle sorte que*

- 1° *il existe une solution optimale $\hat{S}$ de WSTABLE telle que $V_1 \subset \hat{S}$, $V_0 \cap \hat{S} = \emptyset$;*
- 2° *une solution¹⁷ optimale de WSTABLE_r sur le graphe induit par $V_{1/2}$ consiste à assigner la valeur 1/2 à tout sommet ;*
- 3° *V_1 et $V_{1/2}$ n'ont pas de sommets adjacents.*

Ce résultat montre l'intérêt de la classe $\mathcal{C}_{1/2}$ des instances pour lesquelles la solution assignant 1/2 à tout sommet est l'unique solution optimale du programme WSTABLE_r. Elle correspond, en quelque sorte, aux graphes pour lesquels la solution du programme relaxé ne fournit aucune information aidant à la détermination d'une solution de WSTABLE. Plus précisément, $\mathcal{C}_{1/2}$ peut être interprétée comme un noyau d'instances difficiles.

Proposition 26. *WS se réduit au problème restreint à la classe $\mathcal{C}_{1/2}$ avec une expansion préservant tout niveau d'approximation (relativement au rapport d'approximation γ).*

Preuve. La réduction correspondante consiste à construire les ensembles V_1 et $V_{1/2}$ et à compléter toute solution approchée sur le graphe induit par $V_{1/2}$ par l'ensemble V_1 . Les ensembles V_1 et $V_{1/2}$ n'ayant aucun sommet adjacent, la solution est bien réalisable. Par ailleurs, notons respectivement $\lambda_{1/2}$ (λ) la valeur approchée correspondant au graphe induit par $V_{1/2}$ (graphe G) et $\beta_{1/2}$ (β) la valeur optimale dans ce graphe. On a $\lambda = w(V_1) + \lambda_{1/2}$, $\beta = w(V_1) + \beta_{1/2}$, de sorte que le rapport correspondant vérifie $\lambda/\beta \geq \lambda_{1/2}/\beta_{1/2}$, ce qui conclut la preuve. ■

7.5 Famille critique d'instances

Pour conclure, évoquons des travaux en cours qui s'inscrivent dans la continuité de la problématique que nous venons d'étudier. L'idée est toujours de définir une notion de difficulté des instances en vue d'étudier des résultats d'approximation mais, contrairement aux notions discutées dans les paragraphes précédents, ce nouveau point de vue se rapporte à un algorithme ou à

17. La preuve originelle de [51] peut être adaptée pour garantir l'unicité de cette solution.

une classe d'algorithmes. L'objet n'est plus de définir une difficulté intrinsèque de certaines instances mais de réfléchir, étant donné un algorithme, à la structure des instances pour lesquelles il se comporte le moins bien.

Définition 9. ([49]) étant donné un problème Π (d'ensembles d'instances $\mathcal{I}$), un algorithme approché A pour Π et une mesure d'approximation ρ ($\rho \in [0, 1]$ et d'autant plus proche de 1 que l'instance est bien résolue), une famille d'instances $S \subset \mathcal{I}$ est *critique* pour A si, $\forall I_0 \in \mathcal{I}$, $\exists I_1 \in S$, tel que $\rho_A(I_1) \leq \rho_A(I_0)$. ■

Une conséquence immédiate de cette définition est que si l'algorithme A garantit un rapport constant ρ_0 pour les instances S , cette analyse est valable pour toute instance. Cette notion n'est adaptée qu'à l'analyse d'algorithmes à rapport constant, mais il serait facile de l'étendre au cas d'autres niveaux d'approximation. Cependant, ce cadre restreint suffit pour montrer l'esprit de la démarche et son intérêt opérationnel. Remarquons aussi la similarité avec le paragraphe 7.4, mais la différence essentielle est qu'ici la réduction du problème général au sous-problème n'est valable que pour un algorithme. Dans la pratique, la forme particulière de l'algorithme implique une structure des instances correspondantes beaucoup plus riche que dans le cadre de réductions générales. C'est ce qui rend cette notion opérante. Par contre, les instances mises en évidence n'ont pas de signification particulière en termes de difficulté ; elles sont plutôt significatives du comportement de A . Néanmoins, les deux points de vue pourraient être rapprochés par l'étude de notions intermédiaires correspondant à des instances critiques pour des familles d'algorithmes.

Ce cadre restreint étudié dans [49] s'est d'ores et déjà avéré particulièrement fécond. En particulier, sur l'exemple de BINPACKING, nous mettons évidence dans [19] une famille d'instances critiques pour l'algorithme *first-fit-decreasing* et la mesure différentielle. La preuve illustre comment la définition 9 intervient dans la conception de la famille S et met en évidence des propriétés vérifiées par toute famille d'instances critiques pour cet algorithme et cette mesure. Dans un second temps, une simple évaluation de la mesure sur la famille trouvée permet d'établir que l'algorithme *first-fit-decreasing* garantit le rapport différentiel $3/4$.

8 Conclusion

Cet article se veut une tribune sur l'approximation polynomiale dans laquelle nous présentons, avec un regard critique, différents concepts de ce domaine, des exemples les illustrant et les débats qu'ils soulèvent. La plupart de ces concepts sont relativement récents, certains sont même présentés ici dans leur première mouture. Nous relatons donc, en quelque sorte, la naissance (encore en cours) d'un cadre conceptuel pour l'approximation. Pour cela, nous avons choisi d'insister sur la démarche qui a conduit à modifier des concepts à partir de leur analyse critique et sur les possibilités qu'apportent ces nouveautés. Mais au delà de l'évocation d'un nouveau cadre à l'étude, la lecture de cet article est aussi l'occasion de se familiariser avec l'approximation, ses outils, ses enjeux et ses points de vue.

Ainsi, cette discussion s'adresse autant aux non spécialistes de l'approximation qu'aux chercheurs travaillant sur ce thème. Les premiers y trouveront l'occasion d'aborder ce domaine par une approche critique à laquelle ils peuvent accéder grâce aux exemples simples que nous avons sélectionnés. Les seconds y trouveront quelques aspects d'un débat sur la pertinence des concepts de l'approximation, débat auquel nous les convions car il reste d'actualité. Ils pourront également y trouver une présentation de nouveaux concepts assortie de quelques exemples significatifs.

La section 2 est une première introduction à l'approximation polynomiale. Elle présente des concepts clef qui sembleront presque banals à quiconque à déjà travaillé dans ce cadre. Cependant, ces rappels sont déjà l'occasion de poser un regard critique : nous évoquons en particulier la question de la mesure d'approximation qui a conduit à l'étude du rapport différentiel. Notre approche privilégie quelques idées phares mettant en évidence les enjeux de l'approximation ; ceux-ci nous

semblent essentiels pour comprendre les concepts et les motivations qui nous conduisent à les affiner.

La section 3 s'intéresse à une classification « absolue » des problèmes d'optimisation **NPO**. La richesse de cette structure est illustrée par la grande diversité des résultats possibles. L'exemple de STABLE justifie le besoin d'enrichir la gamme usuelle des résultats en introduisant l'idée des chaînes d'approximation qui consistent à étudier globalement le comportement d'une suite d'algorithmes. Cet exemple met également en évidence quelques questions générales qui surviennent fréquemment dans l'étude de problèmes difficiles du point de vue de l'approximation : parmi elles mentionnons le rôle des paramètres de l'instance et la manière dont ils conditionnent sa difficulté. Nous y revenons dans les autres sections. La notion de paramètre est d'ailleurs enrichie par le concept de chaîne d'approximation : il s'agit, selon les cas, de montrer comment évolue la qualité des garanties en fonction de la forme de l'instance ou, à instance fixée, comment évolue la qualité de la solution et la complexité algorithmique en fonction du paramètre de la chaîne.

La section 4 complète la précédente en élargissant la gamme des outils permettant de décrire, dans l'absolu, les possibilités de résolution d'un problème difficile. Plus précisément elle s'intéresse à deux notions de limite en approximation qui offrent de nouvelles possibilités pour l'analyse fine d'algorithmes d'approximation. La première, directement associée au concept de chaîne d'approximation, permet de rendre compte de manière concise du comportement des algorithmes A_k de la chaîne lorsque le paramètre k tend vers l'infini. La qualité de l'approximation augmentant en général avec k , ce comportement limite permet d'évaluer le meilleur type de rapport d'approximation qu'on peut déduire de l'analyse de la chaîne. Cette information est intéressante à confronter à l'analyse en complexité de la chaîne algorithmique. La seconde notion, quant à elle, est celle de *comportement asymptotique*, terme très souvent employé mais rarement défini. Il correspond à une certaine idée de limite des instances. Nous ne prétendons pas apporter une réponse absolue à cette question ; notre objet est plutôt de convaincre de sa pertinence. Il s'agit d'un exemple particulièrement significatif de l'importance qu'il y a, selon nous, à conserver un regard critique sur le type de résultats obtenus indépendamment de leur exactitude mathématique. Ce point nous semble essentiel pour permettre une exploitation des résultats et pour préserver l'intérêt de l'approximation comme outil de compréhension des problèmes. La réponse que nous proposons est clairement orientée et relève d'un choix que nous avons explicité et justifié ; elle consiste à voir le résultat limite comme une simplification du résultat général permettant de mieux appréhender sa teneur sur les instances les plus difficiles. Un autre intérêt de cette étude est justement de soulever la question de la difficulté des instances du point de vue de l'approximation. En effet, la notion d'*ordre de difficulté*, qui nous a permis de définir de manière systématique l'idée d'« instances tendant vers l'infini », représente une notion de difficulté.

Les sections 3 et 4 forment un tout puisqu'elles traitent d'outils permettant de décrire directement (et dans l'absolu) la possibilité d'approximer un problème **NPO**. La section 5 fournit alors quelques exemples significatifs de résultats relatifs à ces deux sections.

La section 6 s'intéresse à une autre manière d'aborder la classification des problèmes de **NPO** en fonction de leurs propriétés d'approximation. Il s'agit cette fois de comparer différents problèmes du point de vue de l'approximation indépendamment des résultats connus pour chacun d'eux. L'outil des réductions permet d'établir un préordre sur **NPO**. Nous avons discuté une nouvelle forme de réductions permettant d'unifier, sous un point de vue déterminé, les multiples notions existantes. Là encore, conformément à nos objectifs, cette discussion offre d'abord aux non-spécialistes l'occasion de se familiariser avec les réductions, d'en comprendre l'esprit sans en subir la haute technicité. En effet, notre définition présente l'avantage de la simplicité et met en avant les objectifs que nous avons choisis pour cet outil. Mais cette définition permet aussi, par sa grande souplesse d'utilisation, d'offrir un outil identique pour tous types de transformations, ce qui s'avère d'une grande richesse pour l'approximation. Cette unification permet notamment de

rendre plus simple la comparaison entre différentes réductions impliquant les mêmes problèmes. Nos exemples montrent la variété des situations pouvant relever de ce cadre.

Ces deux classifications, absolue et relative, sont complémentaires. La comparaison de résultats positifs et négatifs pour différents problèmes permet d'établir l'impossibilité de concevoir certaines réductions. À l'inverse les réductions permettent de transférer des résultats d'un problème à un autre. De nombreux résultats d'approximation sont d'ailleurs le produit de réductions. Le fait que ces résultats sont compétitifs, voire même pour certains non égalés par une méthode directe nous conforte dans l'idée que les réductions en approximation reflètent des liens étroits et significatifs entre les problèmes. Les résultats obtenus montrent la puissance opérationnelle de cet outil; ils illustrent son intérêt pour comprendre les liens entre problèmes ou, pour un même problème, les liens entre ses différentes approximations. Ils peuvent aussi, dans certains cas, nous mettre en garde contre des écueils. L'exemple des STABLE et CLIQUE en est une bonne illustration : une idée répandue associe ces problèmes et les considère comme totalement équivalents. C'est vrai dans le cadre de la résolution exacte ainsi que dans le cas de l'approximation avec des rapports ne dépendant que de l'ordre du graphe, ce qui fut longtemps (et même parfois reste encore) le seul cadre formel pour l'approximation. Or poser la question d'une réduction entre ces problèmes et l'explicitation de son effet sur le rapport d'approximation nous a conduit à les différencier. Toujours par l'outil des réductions nous avons mis en évidence le rôle du paramètre Δ pour leur approximation, ce qui remet en cause (et laisse ouvert) leur équivalence. Une autre conclusion à ce travail est que, non seulement les notions de difficulté, mais aussi celles d'équivalence doivent être associées à un niveau d'approximation ou plus généralement à une certaine qualité de résolution. Tous les problèmes considérés sont équivalents du point de vue de la résolution exacte alors qu'il ne le sont pas du point de vue de l'approximation. Vu la richesse des différents niveaux de résolution offerte par l'approximation, il est alors naturel de considérer que deux problèmes peuvent être équivalents pour un type de résolution et pas pour un autre. L'exemple de STABLE et CLIQUE montre l'intérêt de ce type de question puisque cette démarche a conduit à l'amélioration du rapport d'approximation garanti pour CLIQUE. De telles considérations sont également précieuses pour comprendre les problèmes en question. Notre définition de réduction entièrement paramétrée par le type de résolution est particulièrement appropriée à de telles études.

Enfin, la section 7 aborde la question de la difficulté des instances d'un problème. Elle donne l'occasion d'une synthèse des précédentes sections. En effet, les outils que nous avons discutés pour établir une hiérarchie des problèmes permettent d'étudier la structure des instances d'un problème, c'est à dire la structure interne du problème. Pour cela, il suffit d'associer à des classes d'instances des sous-problèmes du problème initial et d'étudier leur difficulté relative à l'aide de résultats d'approximation et de réductions. On retrouve dans ce contexte la notion d'ordre de difficulté. Ce concept permet d'exprimer l'incidence d'un paramètre de l'instance sur les possibilités de sa résolution. Les nombreux exemples que nous avons décrits dans les sections 4 et 7 montrent la richesse de cette notion et la diversité des situations qu'elle peut représenter. Cette étude est aussi l'occasion de montrer comment l'outil des réductions permet de mettre en évidence une classe d'instances plus difficiles sur lesquelles focaliser son attention. Enfin, nous montrons comment certains travaux récents s'inscrivent dans ce cadre : nous évoquons en particulier le cas de sous-problèmes intéressants de STABLE ainsi que la notion d'instances critiques qui permet de mieux comprendre les limites de certains résultats d'approximation et donne la possibilité de les dépasser.

Enfin, nous tenons à insister une nouvelle fois sur le caractère non figé des réponses que nous avons proposées dans ce document. Nous espérons avoir convaincu de la grande richesse offerte par ce formalisme souple et systématique. Il autorise non seulement une très large palette de résultats d'approximation mais de plus, et c'est peut-être le plus intéressant, il donne accès

à de nombreux points de vue ou questions générales qui nous semblent dignes d'être étudiées systématiquement pour l'ensemble des problèmes de **NPO**. Parmi ces questions générales figurent notamment le rôle des paramètres de l'instance ou encore les liens entre les versions pondérée et non-pondérée de différents problèmes. Ce cadre ouvre également la voie à l'étude de nouvelles mesures d'approximation complémentaires aux rapports classique et différentiel : la volonté de pouvoir traiter ces deux rapports de front en paramétrant les résultats et les outils par le type de mesure permet immédiatement d'inclure, en tant que paramètre, de nouvelles mesures. La richesse et la complémentarité entre résultats classiques et différentiels ne peut que nous encourager à étudier d'autres mesures ; pour cela aucune modification du formalisme ne sera nécessaire.

Aussi, ce travail sera-t-il amené à être poursuivi et enrichi. Notre principal souci a été de le rendre compatible avec ces prochaines évolutions en fonction des nouveaux besoins, points de vue et résultats du domaine.

Références

- [1] ALFANDARI, L. *Approximation de problèmes de couverture et de partitionnement de graphes*. Ph.D. thesis, LAMSADE, Université Paris-Dauphine, 1999.
- [2] ALON, N. and KAHALE, N. Approximating the independence number via the θ -function. *Math. Programming*, 1998.
- [3] ARORA, S., LUND, C., MOTWANI, R., SUDAN, M., and SZEGEDY, M. Proof verification and intractability of approximation problems. In *Proc. FOCS'92*, pages 14–23. 1992.
- [4] AUSIELLO, G., CRESCENZI, P., and PROTASI, M. Approximate solutions of NP optimization problems. *Theoret. Comput. Sci.*, 150:pages 1–55, 1995.
- [5] AUSIELLO, G., D'ATRI, A., and PROTASI, M. Structure preserving reductions among convex optimization problems. *J. Comput. System Sci.*, 21:pages 136–153, 1980.
- [6] BERGE, C. *Graphs and hypergraphs*. North Holland, Amsterdam, 1973.
- [7] BERMAN, P. and FÜRER, M. Approximating maximum independent set in bounded degree graphs. In *Proc. Symposium on Discrete Algorithms*, pages 365–371. 1994.
- [8] BERMAN, P. and HARTMANIS, J. On isomorphisms and density of NP and other complete sets. *SIAM J. Comput.*, 6(2):pages 305–322, 1977.
- [9] BÖCKENHAUER, H.-J., HROMKOVIČ, J., KLASING, R., SEIBERT, S., and UNGER, W. Towards the notion of stability of approximation algorithms and the traveling salesman problem. Report 31, Electr. Colloq. Computational Comp., 1999.
- [10] BOPPANA, B. B. and HALLDÓRSSON, M. M. Approximating maximum independent sets by excluding subgraphs. *BIT*, 32(2):pages 180–196, 1992.
- [11] CHVÁTAL, V. A greedy-heuristic for the set covering problem. *Math. Oper. Res.*, 4:pages 233–235, 1979.
- [12] COOK, S. A. The complexity of theorem-proving procedures. In *Proc. STOC'71*, pages 151–158. 1971.
- [13] CREIGNOU, N. *Temps linéaire et problèmes NP-complets*. Ph.D. thesis, Université de Caen, 1993.
- [14] CRESCENZI, P. A short guide to approximation preserving reductions. In *Proc. Conference on Computational Complexity*, pages 262–273. 1997.
- [15] CRESCENZI, P., KANN, V., SILVESTRI, R., and TREVISAN, L. Structure in approximation classes. Technical Report TR96-066, Electronic Colloquium on Computational Complexity, 1996. Available on [www_address: http://www.eccc.uni-trier.de/eccc/](http://www.eccc.uni-trier.de/eccc/).
- [16] CRESCENZI, P. and PANCONESI, A. Completeness in approximation classes. *SIAM J. Comput.*, 1991.

- [17] DEMANGE, M., GRISONI, P., and PASCHOS, V. TH. Differential approximation algorithms for some combinatorial optimization problems. *Theoret. Comput. Sci.*, 209:pages 107–122, 1998.
- [18] DEMANGE, M., MONNOT, J., and PASCHOS, V. TH. Bridging gap between standard and differential polynomial approximation: the case of bin-packing. *Appl. Math. Lett.*, 12:pages 127–133, 1999.
- [19] DEMANGE, M., MONNOT, J., and PASCHOS, V. TH. Maximizing the number of unused bins. *Found. Comput. Decision Sci.*, 26(2):pages 169–186, 2001.
- [20] DEMANGE, M. and PASCHOS, V. TH. On an approximation measure founded on the links between optimization and polynomial approximation theory. *Theoret. Comput. Sci.*, 158:pages 117–141, 1996.
- [21] DEMANGE, M. and PASCHOS, V. TH. Valeurs extrémales d'un problème d'optimisation combinatoire et approximation polynomiale. *Math. Inf. Sci. Humaines*, 135:pages 51–66, 1996.
- [22] DEMANGE, M. and PASCHOS, V. TH. Improved approximations for maximum independent set via approximation chains. *Appl. Math. Lett.*, 10(3):pages 105–110, 1997.
- [23] DEMANGE, M. and PASCHOS, V. TH. Towards a general formal framework for polynomial approximation. Cahier du LAMSADE 177, LAMSADE, Université Paris-Dauphine, 2001.
- [24] DOWNEY, R. G. and FELLOWS, M. R. *Parameterized complexity*. Monographs in Computer Science. Springer, New York, 1999.
- [25] DUH, R. and FÜRER, M. Approximation of k -set cover by semi-local optimization. In *Proc. STOC'97*, pages 256–265. 1997.
- [26] FEIGE, U. and KILIAN, J. Zero Knowledge and the chromatic number. In *Proc. Conference on Computational Complexity*, pages 278–287. 1996.
- [27] FERNANDEZ DE LA VEGA, W. Sur la cardinalité maximum des couplages d'hypergraphes aléatoires uniformes. *Discrete Math.*, 40:pages 315–318, 1982.
- [28] GAREY, M. R. and JOHNSON, D. S. *Computers and intractability. A guide to the theory of NP-completeness*. W. H. Freeman, San Francisco, 1979.
- [29] HALLDÓRSSON, M. M. Approximating the minimum maximal independence number. *Inform. Process. Lett.*, 46:pages 169–172, 1993.
- [30] HALLDÓRSSON, M. M. A still better performance guarantee for approximate graph coloring. *Inform. Process. Lett.*, 45(1):pages 19–23, 1993.
- [31] HALLDÓRSSON, M. M. Approximations via partitioning. JAIST Research Report IS-RR-95-0003F, Japan Advanced Institute of Science and Technology, Japan, 1995.
- [32] HALLDÓRSSON, M. M. Approximating k -set cover and complementary graph coloring. In *Proc. International Integer Programming and Combinatorial Optimization Conference*, number 1084 in Lecture Notes in Computer Science, pages 118–131. Springer Verlag, 1996.
- [33] HALLDÓRSSON, M. M. and RADHAKRISHNAN, J. Greed is good: approximating independent sets in sparse and bounded-degree graphs. In *Proc. STOC'94*, pages 439–448. 1994.
- [34] HALLDÓRSSON, M. M. and RADHAKRISHNAN, J. Improved approximations of independent sets in bounded-degree graphs via subgraph removal. *Nordic Journal of Computing*, 1(4):pages 475–492, 1994.
- [35] HASSIN, R. and LAHAV, S. Maximizing the number of unused colors in the vertex coloring problem. *Inform. Process. Lett.*, 52:pages 87–90, 1994.
- [36] HÅSTAD, J. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:pages 105–142, 1999.

- [37] HOCHBAUM, D. S. Efficient bounds for the stable set, vertex cover and set packing problems. *Discrete Appl. Math.*, 6:pages 243–254, 1983.
- [38] HRONKOVIC, J. *Algorithmics for hard problems*. Springer, 2001.
- [39] IBARRA, O. H. and KIM, C. E. Fast approximation algorithms for the knapsack and sum of subset problems. *J. Assoc. Comput. Mach.*, 22(4):pages 463–468, 1975.
- [40] JOHNSON, D. S. Approximation algorithms for combinatorial problems. *J. Comput. System Sci.*, 9:pages 256–278, 1974.
- [41] KANN, V. Polynomially bounded problems that are hard to approximate. *Nordic J. Computing*, 1:pages 317–331, 1994.
- [42] KARGER, D., MOTWANI, R., and SUDAN, M. Approximate graph coloring by semidefinite programming. *J. Assoc. Comput. Mach.*, 45(2):pages 246–265, 1998.
- [43] KARP, R. M. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of computer computations*, pages 85–103. Plenum Press, New York, 1972.
- [44] KHANNA, S., MOTWANI, R., SUDAN, M., and VAZIRANI, U. On syntactic versus computational views of approximability. *SIAM J. Comput.*, 28:pages 164–191, 1998.
- [45] LEWIS, H. R. and PAPADIMITRIOU, C. H. *Elements of the theory of computation*. Prentice-Hall, 1981.
- [46] LORENZO, J. Approximation des solutions et des valeurs des problèmes NP-complets. Thèse de Doctorat, CERMSEM, Université Paris I. En préparation.
- [47] LUND, C. and YANNAKAKIS, M. On the hardness of approximating minimization problems. *J. Assoc. Comput. Mach.*, 41(5):pages 960–981, 1994.
- [48] LYNCH, N. and LIPTON, J. On structure preserving reductions. *SIAM J. Comput.*, 7(2):pages 119–126, 1978.
- [49] MONNOT, J. *Familles critiques d'instances et approximation polynomiale*. Ph.D. thesis, LAMSADE, Université Paris-Dauphine, 1998.
- [50] MOTWANI, R. *Lecture notes on approximation algorithms – Volume I*. Stanford University, 1993.
- [51] NEMHAUSER, G. L., WOLSEY, L. A., and FISCHER, M. L. An analysis of approximations for maximizing submodular set functions. *Math. Programming*, 14:pages 265–294, 1978.
- [52] ORPONEN, P. and MANNILA, H. On approximation preserving reductions: complete problems and robust measures. Technical Report C-1987-28, Dept. of Computer Science, University of Helsinki, Finland, 1987.
- [53] PAPADIMITRIOU, C. H. and STEIGLITZ, K. *Combinatorial optimization: algorithms and complexity*. Prentice Hall, New Jersey, 1981.
- [54] PAPADIMITRIOU, C. H. and YANNAKAKIS, M. Optimization, approximation and complexity classes. *J. Comput. System Sci.*, 43:pages 425–440, 1991.
- [55] PAZ, A. and MORAN, S. Non deterministic polynomial optimization problems and their approximations. *Theoret. Comput. Sci.*, 15:pages 251–277, 1981.
- [56] RAZ, R. and SAFRA, S. A sub-constant error probability low-degree test and a sub-constant error probability PCP characterization of NP. In *Proc. STOC'97*, pages 475–484. 1997.
- [57] SIMCHI-LEVI, D. New worst-case results for the bin-packing problem. *Naval Res. Logistics*, 41:pages 579–585, 1994.
- [58] SIMON, H. U. On approximate solutions for combinatorial optimization problems. *SIAM J. Disc. Math.*, 3(2):pages 294–310, 1990.
- [59] TURÁN, P. On an extremal problem in graph theory (in Hungarian). *Mat. Fiz. Lapok*, 48:pages 436–452, 1941.

Annexes

A Quelques problèmes NPO

Stable maximum (STABLE)

INSTANCE : graphe $G = (V, E)$;

SOLUTION RÉALISABLE : un ensemble stable, *i.e.*, un ensemble $V' \subseteq V$ tel que $\forall (v, v') \in V' \times V', vv' \notin E$;

OBJECTIF : maximiser le cardinal de V' , *i.e.*, $|V'|$;

VERSION PONDÉRÉE (WSTABLE) : chaque sommet $v \in V$ a une valeur $w_v > 0$ et l'objectif consiste à maximiser $w(V') = \sum_{v \in V'} w_v$; PWSTABLE et BWSTABLE désignent les restrictions de WSTABLE, respectivement au cas de poids polynomiaux et de poids bornés par une constante.

Clique maximum (CLIQUE)

INSTANCE : graphe $G = (V, E)$;

SOLUTION RÉALISABLE : une clique *i.e.*, un ensemble $V' \subseteq V$ tel que $\forall (v, v') \in V' \times V', vv' \in E$;

OBJECTIF : maximiser le cardinal de V' ;

VERSION PONDÉRÉE (WCLIQUE) : chaque sommet $v \in V$ a une valeur $w_v > 0$, et l'objectif consiste à maximiser $w(V') = \sum_{v \in V'} w_v$.

Sous-graphe k -colorable maximum (kCl)

INSTANCE : graphe $G = (V, E)$;

SOLUTION RÉALISABLE : ensemble de sommets V' tel que $G[V']$ est k -colorable ;

OBJECTIF : maximiser le cardinal de V' ;

VERSION PONDÉRÉE ($WkCl$) : chaque sommet $v \in V$ a une valeur $w_v > 0$, et l'objectif consiste à maximiser $w(V') = \sum_{v \in V'} w_v$.

Couverture minimum de sommets¹⁸ (G-TRANSVERSAL)

INSTANCE : graphe $G = (V, E)$;

SOLUTION RÉALISABLE : une couverture de sommets, *i.e.*, un ensemble $V' \subseteq V$ tel que $\forall vv' \in E, \{v, v'\} \cap V' \neq \emptyset$;

OBJECTIF : minimiser le cardinal de V' .

Coloration minimum des sommets (COLORATION)

INSTANCE : graphe $G = (V, E)$;

SOLUTION RÉALISABLE : une coloration de G , *i.e.*, une partition de V en ensembles stables disjoints non vides $(V_1, \dots, V_p)$;

OBJECTIF : minimiser le nombre p de stables V_i ;

COMMENTAIRE : chaque stable correspond à une couleur, il s'agit alors de colorer les sommets avec un nombre minimum de couleurs de sorte que deux sommets adjacents n'ont pas la même couleur.

Coloration minimum des arêtes

INSTANCE : graphe $G = (V, E)$;

SOLUTION RÉALISABLE : une coloration des arêtes de G , *i.e.*, une partition de E en couplages disjoints non-vides, $(M_1, \dots, M_p)$;

OBJECTIF : minimiser le nombre p de couplages M_i ;

COMMENTAIRE : il s'agit du problème de coloration sur « le line-graph » de G .

Stable maximal minimum ou Stable dominant minimum (STABLEMAXMIN)

INSTANCE : graphe $G = (V, E)$;

SOLUTION RÉALISABLE : un stable maximal pour l'inclusion V' (*i.e.*, pour tout sommet

18. Ou **Transversal minimum d'un graphe** ([6]).

$v \notin V', V' \cup \{v\}$ non stable);
 OBJECTIF : minimiser le cardinal de V' ;
 VERSION PONDÉRÉE (WSTABLEMAXMIN) : chaque sommet $v \in V$ a une valeur $w_v > 0$ et l'objectif consiste à minimiser $w(V') = \sum_{v \in V'} w_v$; la version PWSTABLEMAXMIN est définie comme celle de STABLE.

Arbre de STEINER

INSTANCE : graphe complet $G = (V, E)$ dont chaque arête $e \in E$ a une valeur $w_e > 0$, un ensemble $R \subset V$;
 SOLUTION RÉALISABLE : un sous-graphe partiel (V', E') de G qui est un arbre et tel que $R \subset V'$;
 OBJECTIF : minimiser la valeur de l'arbre, *i.e.*, $\sum_{e \in E'} w_e$.

Voyageur de commerce (TSP)

INSTANCE : graphe complet $G = (V, E)$ dont chaque arête $e \in E$ a une valeur $w_e > 0$;
 SOLUTION RÉALISABLE : un cycle hamiltonien $H \subset E$ (graphe partiel connexe tel que tout sommet a un degré 2);
 OBJECTIF : minimiser la valeur du cycle, *i.e.*, $\sum_{e \in H} w_e$;
 COMMENTAIRE : Δ -TSP est la version de TSP où les distances sur les arêtes vérifient les inégalités triangulaires.

Couverture d'ensemble¹⁹ (H-TRANSVERSAL)

INSTANCE : un ensemble de base E et une collection $\mathcal{S}$ de parties de E couvrant E , *i.e.*, $\mathcal{S} \subset 2^E$, $\cup_{S \in \mathcal{S}} S = E$;
 SOLUTION RÉALISABLE : une partie $\mathcal{S}' \subseteq \mathcal{S}$ couvrant E , *i.e.*, $\cup_{S \in \mathcal{S}'} S = E$;
 OBJECTIF : minimiser le cardinal de $\mathcal{S}'$;
 COMMENTAIRE : H-3-TRANSVERSAL est la version où $\forall S \in \mathcal{S}$, $|S| \leq 3$.

3-couplage maximum

INSTANCE : ensemble $M \subset A \times B \times C$ où A , B , et C sont disjoints et de même cardinal q ;
 SOLUTION RÉALISABLE : 3-couplage, *i.e.*, $M' \subseteq M$ tel que les triplets de M' n'ont, 2 à 2, aucune coordonnée en commun;
 OBJECTIF : maximiser le cardinal du couplage;
 COMMENTAIRE : 3-DM est le problème de décision associé : pour chaque instance, la question est : *existe-t-il un 3-couplage de taille q ?* ; 3-DM est **NP**-complet.

Bin-packing (BINPACKING)

INSTANCE : une liste de rationnels $L = (x_1, \dots, x_n)$, $\forall i \in \{1, \dots, n\}$, $0 < x_i \leq 1$;
 SOLUTION RÉALISABLE : une partition de L en ensembles (appelés *bins*) $B_1, \dots, B_p$ de sorte que la somme des nombres de chaque *bin* ne dépasse pas 1;
 OBJECTIF : minimiser le nombre p de *bins*;
 COMMENTAIRE : il s'agit donc de ranger les nombres en un nombre minimum de boîtes de capacité 1.

Sac à dos (Ks)

INSTANCE : deux ensembles de n rationnels $\{a_1, \dots, a_n\}$ et $\{b_1, \dots, b_n\}$, un rationnel B ;
 SOLUTION RÉALISABLE : ensemble d'indices $S \subseteq \{1, \dots, n\}$ tel que $\sum_{i \in S} b_i \leq B$;
 OBJECTIF : maximiser $\sum_{i \in S} a_i$;
 COMMENTAIRE : il s'agit de la programmation linéaire en variables bivalentes et à une contrainte ; le problème existe aussi en version minimisation.

Couplage maximum

INSTANCE : graphe $G = (V, E)$;
 SOLUTION RÉALISABLE : un couplage, *i.e.*, $E' \subseteq E$ tel que les arêtes de E' sont 2 à 2 non

19. Ou **Transversal d'un hypergraphe** ([6]).

adjacentes ;
 OBJECTIF : maximiser $|E'|$;
 COMMENTAIRE : le problème est polynomial ([53]).

Couverture minimum d'arêtes

INSTANCE : graphe $G = (V, E)$;
 SOLUTION RÉALISABLE : une couverture d'arêtes, *i.e.*, $E' \subseteq E$ tel que $\forall v \in V, \exists u \in V, uv \in E'$ (ensemble d'arêtes touchant tous les sommets) ;
 OBJECTIF : minimiser $|E'|$;
 COMMENTAIRE : résolu polynomialement par un algorithme de couplage.

Satisfiabilité pondérée maximum

INSTANCE : ensemble de variables booléennes X , ensemble C de clauses disjonctives sur X , chaque clause c_i est munie d'une valeur v_i ;
 SOLUTION RÉALISABLE : affectation de vérité aux variables ;
 OBJECTIF : maximiser la valeur des clauses vraies *i.e.*, $\sum_i v_i t(c_i)$ où $t(c_i)$ vaut 1 si c_i est satisfaite, 0 sinon.

B Rappel de quelques définitions de base

Un algorithme polynomial approché est un algorithme qui détermine, pour chaque instance I d'un problème $\Pi \in \mathbf{NPO}$, une solution réalisable en temps polynomial en la taille de I . Il garantit un rapport $\rho(I)$ si cette solution vérifie, pour toute instance I de Π :

- 1° si on travaille avec le rapport γ (approximation classique ou standard)
 - 1° $\lambda(I)/\beta(I) \geq \rho(I)$, si Π est un problème de maximization, ou
 - 2° $\beta(I)/\lambda(I) \geq \rho(I)$, si Π est un problème de minimization
- 2° $|\omega(I) - \lambda(I)|/|\omega(I) - \beta(I)| \geq \rho(I)$ pour le cas de l'approximation différentielle (rapport δ).

Nous pouvons classer les algorithmes en fonction du meilleur rapport (connus) qu'ils garantissent ; c'est ce que nous appelons dans l'article niveau d'approximation. Nous donnons ci-après une liste des classes mentionnées dans cet article et dans la littérature du domaine :

Algorithme à rapport constant ρ :

quand $\rho(I)$ est une constante ne dépendant pas de I , *i.e.*, ne dépendant d'aucun paramètre de l'instance du problème ;

Schéma d'approximation polynomial :

suite d'algorithmes indicée par $\epsilon > 0$ garantissant le rapport $1 - \epsilon$; chaque algorithme est polynomial ;

Schéma complet d'approximation polynomial :

schéma d'approximation dont la complexité est polynomial en $|I|$ et en $1/\epsilon$;

Algorithmes à rapport logarithmique :

cas où $\rho(I) \geq O(1/\log |I|)$;

Algorithmes à rapport $O(n^{\epsilon-1})$:

cas où $\exists \epsilon > 0$ tel que $\rho(I) \geq O(n^{\epsilon-1})$.

La classification considérée pour les algorithmes approchés peut être étendue aux problèmes si on considère que le rapport d'approximation d'un problème Π est le rapport du meilleur algorithme résolvant Π . Ainsi, on peut considérer une structure pour $\mathbf{NPO}$ par rapport à l'approximabilité de ses problèmes. La figure 1 représente les classes d'approximabilité les plus usuelles, évidemment sous la conjecture $\mathbf{P} \neq \mathbf{NP}$. Par analogie, nous notons $\mathbf{PO}$ la classe des problèmes d'optimisation polynomiaux. Certaines classes contiennent même des problèmes complets pour des réductions

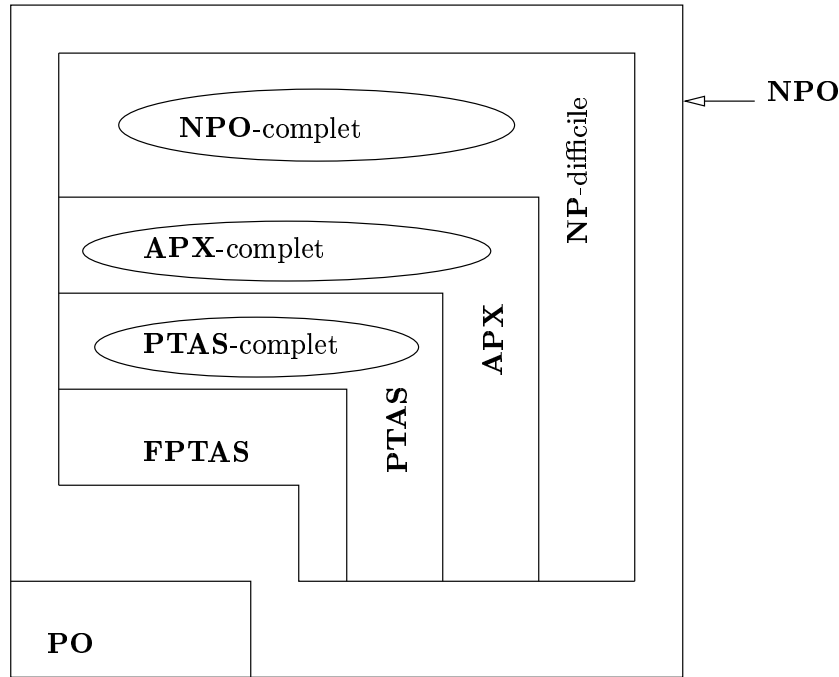


FIG. 1 – *Le monde de l'approximabilité selon les conjectures des chercheurs.*

appropriées. Nous rappelons qu'étant donné une classe $\mathbf{C}$ et une réduction R , un problème Π est $\mathbf{C}$ -complet si

1° $\Pi \in \mathbf{C}$ et

2° pour tout problème $\Pi' \in \mathbf{C}$, Π' se R -réduit à Π .

Un problème **NPO-complet** est tel que s'il admettait un algorithme à rapport constant, alors tous les autres problèmes **NPO-complets** seraient approximables à rapport constant. Notons que $\mathbf{PO} = \mathbf{NPO} \iff \mathbf{P} = \mathbf{NP}$.

Nous donnons ci-après les principales classes d'approximabilité pour l'approximation classique. Évidemment cette classification reste valable pour l'approximation différentielle à ceci près que nous ne disposons pas, pour le moment, de réductions appropriées pour prouver les différentes complétudes ; justement la réduction $\mathbf{FP}$ nous semble adéquate pour continuer dans cette voie.

- APX :** la classe des problèmes **NPO** admettant un algorithme à rapport constant ; la classe **APX-complet** contient les problèmes **APX**, complets sous la réduction $\mathbf{A}$; elle est connue comme étant non-vide ;
- PTAS :** la classe des problèmes **NPO** admettant un schéma d'approximation polynomial ; la classe **PTAS-complet** est la classe des problèmes **PTAS**, complets sous la réduction $\mathbf{P}$;
- FPTAS :** la classe des problèmes **NPO** admettant un schéma complet d'approximation polynomial ;
- Log-APX :** la classe des problèmes **NPO** admettant un algorithme approché à rapport logarithmique en I ; l'existence des problèmes **Log-APX-complets** est un problème ouvert intéressant ; une réduction $\mathbf{l-P}$ du type « maître-esclave » (paragraphe 6.5.2) nous semble une piste intéressante pour la preuve d'une telle existence ;
- Poly-APX :** la classe des problèmes **NPO** admettant un algorithme approché garantissant un

rapport qui est un polynôme en la taille de l'instance ; la classe **Poly-APX**-complet est la classe des problèmes **Poly-APX**, complets sous la réduction E ; elle est connue comme étant non-vide.

Il est bien-connu qu'une façon simple de penser à la complétude par rapport à une classe et de se dire qu'un problème Π est **C**-complet signifie qu'il appartient à **C** mais ne peut pas appartenir (sous une hypothèse de complexité) à une classe supérieure. Pour une classe d'approximabilité, ceci signifie qu'il est résoluble avec des rapports d'approximation correspondant à son appartenance à la classe **C** (par exemple, à rapport constant si **C** est **APX**) mais ne peut pas être résolu avec des rapports correspondant à la classe supérieure (par exemple, par un schéma d'approximation polynomial dans notre exemple). La classe **FPTAS** est considérée comme la classe de résolution approchée la plus favorable (sous la conjecture $\mathbf{P} \neq \mathbf{NP}$). Elle n'a pas de problème complet puisque le niveau supérieur de résolution pour ses membres serait la polynomialité. Remarquons enfin que comme la valeur d'un rapport d'approximation appartient à $\mathbb{R}^+$, il existe tout un *continuum* de classes d'approximabilité.