

AUTOUR DE NOUVELLES NOTIONS POUR L'ANALYSE
DES ALGORITHMES D'APPROXIMATION :
DE LA STRUCTURE DE NPO À LA STRUCTURE
DES INSTANCES

MARC DEMANGE¹ ET VANGELIS PASCHOS²

Communiqué par Pierre Tolla

Abstract. This paper is the continuation of the paper “*Autour de nouvelles notions pour l’analyse des algorithmes d’approximation: Formalisme unifié et classes d’approximation*” where a new formalism for polynomial approximation and its basic tools allowing an “absolute” (individual) evaluation the approximability properties of **NP**-hard problems have been presented and discussed. In order to be used for exhibiting a structure for the class **NPO** (the optimization problems of **NP**), these tools must be enriched with an “instrument” allowing comparisons between approximability properties of different problems (these comparisons must be independent on any specific approximation result of the problems concerned). This instrument is the approximability-preserving reductions. We show how to integrate them in the formalism presented and propose the definition of a new reduction unifying, under a specific point of view a great number of existing ones. This new reduction allows also to widen the use of the reductions, restricted until now either between versions of a problem, or between problems, in order to enhance structural relations between frameworks. They allow, for example, to study how standard-approximation properties of a problem transform into differential-approximation ones (for the same problem, or for another one). Finally, we apply the several concepts introduced to the study of the structure (and hardness) of the instances of a problem. This point of view seems particularly fruitful for a better apprehension of the hardness of certain problems and of the mechanisms for the design of efficient solutions for them.

Reçu en novembre 2001.

¹ ESSEC, Cergy-Pontoise, France ; e-mail : demange@essec.fr

© EDP Sciences 2003

² LAMSADE, Université Paris-Dauphine, France ; e-mail : paschos@lamsade.dauphine.fr

Résumé. Cet article est la suite de l'article « *Autour de nouvelles notions pour l'analyse des algorithmes d'approximation : formalisme unifié et classes d'approximation* » où nous avons présenté et discuté, dans le cadre d'un nouveau formalisme pour l'approximation polynomiale (algorithmique polynomiale à garanties de performances pour des problèmes **NP**-difficiles), des outils permettant d'évaluer, dans l'absolu, les propriétés d'approximation de problèmes difficiles. Afin de répondre pleinement à l'objectif de l'approximation polynomiale de mettre en évidence et étudier une structure de la classe **NPO** (problèmes d'optimisation de **NP**), ces outils ont besoin d'être complétés pour offrir la possibilité de comparer, indépendamment de tout résultat spécifique, les propriétés d'approximation de problèmes différents. C'est l'objet de la notion de réduction en approximation à laquelle nous nous intéressons ici. Nous montrons comment l'intégrer au nouveau formalisme et présentons une définition qui unifie, sous un point de vue spécifique, les nombreuses notions existantes. Cette définition permet aussi un emploi systématique de réductions pour étudier des liens entre différents problèmes, entre différentes versions d'un problème ou encore entre le cadre de l'approximation classique et celui de l'approximation différentielle. Comme dans l'article « *Autour de nouvelles notions pour l'analyse des algorithmes d'approximation : formalisme unifié et classes d'approximation* », ce travail est illustré par de nombreux exemples et s'adresse tant aux spécialistes du domaine, pour un débat commun, qu'aux non spécialistes qui ont une occasion de se familiariser avec ce domaine. Enfin, nous appliquons les différents concepts à l'étude de la structure (et la difficulté) des instances d'un problème. Cette problématique s'avère très intéressante pour une meilleure compréhension de la difficulté de certains problèmes et pour leur résolution efficace.

Mots Clés. Complexité, difficulté intrinsèque, analyse des algorithmes et des problèmes, algorithmes d'approximation.

Classification Mathématique. 68Q15, 68Q17, 68Q25, 68W25.

1. INTRODUCTION

Dans l'article [24], nous avons présenté les bases d'un formalisme pour l'approximation polynomiale permettant d'ouvrir son cadre usuel à de nouvelles problématiques. Ce travail avait également pour vocation d'unifier différents aspects de l'approximation en un cadre unique. Il se voulait par ailleurs une analyse critique des concepts utilisés en approximation, analyse permettant d'en comprendre les enjeux et les faiblesses. Pour plus de renseignements sur l'état de l'art de l'approximation polynomiale le/la lecteur/ice intéressé/e est invité/e à consulter les ouvrages [4, 30, 45].

L'objet de l'approximation polynomiale est d'étudier dans quelle mesure un algorithme polynomial peut déterminer, pour toute instance d'un problème d'optimisation **NP**-difficile, une solution réalisable garantissant un certain niveau de qualité (niveau caractérisé par un *rapport d'approximation*). Pour chaque instance du problème, le rapport d'approximation permet d'évaluer la qualité de la solution construite par l'algorithme par rapport à une solution optimale et attribue une valeur entre 0 et 1. L'analyse de la qualité d'approximation d'un algorithme polynomial correspond à un seuil minimum pour la valeur du rapport d'approximation qui est garanti pour toute instance. Pour l'unité et la lisibilité de ce document, nous rappelons en annexe B les principales définitions de [24]. Nous définissons par exemple deux rapports d'approximation présentés dans [24] comme complémentaires. Le premier (dit « classique »), compare la valeur de la solution trouvée par l'algorithme (solution approchée) à la valeur optimale de l'instance. Le second, appelé « rapport différentiel », correspond à la position de la valeur de la solution approchée dans l'intervalle entre les valeurs réalisables extrêmes. Toutes les notions définies dans [24] peuvent être adaptées à l'un ou l'autre des rapports (ou à tout autre d'ailleurs). Nous présentons également la notion de *chaîne polynomiale d'approximation* qui correspond à analyser globalement une suite d'algorithmes et donne lieu à une notion de convergence. Nous rappelons aussi les définitions de degré de difficulté et de *rapport asymptotique* qui sont longuement discutées dans [24].

L'approximation polynomiale et sa panoplie d'outils pour mesurer la qualité d'approximation d'algorithmes permettent de structurer la classe **NPO** des problèmes d'optimisation de **NP**³ en fonction des garanties possibles pour chaque problème. Les *niveaux d'approximation* (annexe B) regroupent les problèmes pour lesquels certains résultats d'approximations sont possibles. D'ailleurs, structurer **NPO** constitue l'un de des principaux enjeux de ce domaine.

Les notions présentées et discutées dans [24] permettent essentiellement de décrire les possibilités d'approximation de problèmes spécifiques et offrent un cadre pour la conception et l'analyse des algorithmes d'approximation. Cette approche « opérationnelle » est essentielle pour l'étude de chaque problème et répond au second objectif de l'approximation qui est la résolution efficace de problèmes spécifiques. Néanmoins, du point de vue structurel, ce cadre ne suffit pas comme nous l'illustrons dans le paragraphe 2. Nous mettons en évidence la nécessité, pour répondre à l'enjeu de structurer **NPO**, d'un outil permettant de comparer différents problèmes du point de vue de l'approximation. Les réductions en approximation ont cette vocation.

Dans la section 2 nous discutons cette notion avec le même état d'esprit que dans l'article [24]. En particulier, cette présentation et les exemples qui y sont donnés peuvent être abordés comme une introduction à ce concept. Du point de vue du spécialiste d'approximation maintenant, cette partie met en évidence comment adapter les réductions au cadre unifié évoqué dans [24] et surtout comment les

³Le cadre d'étude de l'approximation polynomiale.

exploiter pour envisager de manière systématique toute question traitant de la difficulté relative de problématiques ou de cadres.

Dans le paragraphe 2.2, nous proposons un rapide aperçu des principales notions de réductions existantes et mettons en évidence une certaine logique dans l'évolution de ce concept. Elle nous conduira à une nouvelle réduction, notée FP, dont l'objet est d'unifier, 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 et met en évidence dans quel cadre les autres définitions gardent leur intérêt. Les réductions FP s'avèrent être particulièrement bien adaptées au cadre unifié présenté dans [24]. Nous donnons également quelques pistes pour des généralisations ultérieures. Dans le paragraphe 2.5 nous proposons des exemples d'utilisation de ce concept : nous illustrons notamment la possibilité de comparer différents paramètres en fonction desquels exprimer les rapport d'approximation, proposons un exemple de liens entre les approximations classique et différentielle ou encore évoquons la difficulté relative des versions pondérée et non pondérée d'un problème. Enfin, dans la section 3, nous envisageons une problématique naturelle dans le cadre de résultats structurels : au lieu de poser la question de la structure de l'ensemble des problèmes, nous considérons celle de la structure des instances d'un même problème. Il s'agit d'un point de vue complémentaire dont l'objectif est de mieux comprendre ce qui fait la difficulté d'un problème. Nous montrons comment cette question est liée aux concepts de l'approximation décrits dans [24] et dans la première section de cet article. Nous montrons notamment le lien entre ce point de vue, la notion d'ordre de difficulté (annexe 4) et celle de réductions. Nous comparons alors deux manières d'aborder cette question qui sont le reflet des deux approches – statique et relative – observées pour l'étude de la structure de **NPO**. Les différents exemples développés mettent en évidence l'intérêt opérationnel de cette problématique ainsi que les liens intéressants qu'elle entretient avec le cadre classique de l'approximation. Autant d'arguments qui plaident en faveur d'une étude systématique et conjointe de ces différents points de vue.

2. COMPARER LES PROBLÈMES : UN COMPLÈMENT POUR STRUCTURER **NPO**

2.1. MOTIVATIONS

Un des grands enjeux de l'approximation est la définition d'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 prises en compte pour qu'il ne soit pas fait un mauvais usage de telles analyses. Enrichir les outils et les points de vue est un premier palliatif à ces imperfections. Rappelons, dans un même ordre d'idées, les analyses différentielles évoquées dans [24]. Elles 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 [25], est tout à fait adapté à des analyses différentielles (COLORATION $\in \mathbf{D} - \mathbf{APX}$ [19]. 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 aussi 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 [26] 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 ».

2.2. 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. Il montre la logique générale des définitions existantes et permet d'introduire la notion de réduction **FP**.

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 [34] suppose en outre une hypothèse de complexité polynomiale. 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 [31]) la possibilité de réductions plus fines permettant de transférer des résultats d'approximation. Les premiers exemples sont apparus vers 1977 [6] puis se sont considérablement développés pour conduire, aujourd'hui, à une dizaine de notions de réductions, notamment L, LINÉAIRE⁴, CONTINUE, STRICTE APX, PTAS, E, S, AP (un bref descriptif formel de ces réductions peut-être trouvé dans [16]).

Les réductions polynomiales ont d'abord été affinées de manière à préserver la « structure » des instances [6, 9, 37] avant que ne soit mis en évidence, dans [6], 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 I-I) 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 $h \circ A \circ f^5$, 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 [44] et linéaires [42], 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 [40] et la réduction E [35], la définition explicite le rapport d'approximation de $h_{I_1}(y)$ en fonction du rapport pour y . Enfin (notamment pour les réductions A [40], P [18], APX, PTAS [5] ou encore AP [17]), 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 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

⁴À ne pas confondre avec l'équivalence linéaire de [15].

⁵Nous désignons ainsi $I \mapsto h_I \circ A \circ f(I)$.

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, sous certaines conditions, les rapports logarithmiques, polynomiaux et de la forme $1/o(n^{1-\epsilon})$, mais pas ceux de la forme $O(1/n^{1-\epsilon})$. La notion d'*amplification* introduite par Kann [32] 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.

2.3. 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 **I-I**; 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** [44] pour lesquelles 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 1. Soient $(\Pi_1, \Pi_2) \in \mathbf{NPO} \times \mathbf{NPO}$. Une réduction **FP** de Π_1 à Π_2 est un triplet $\alpha_{\Pi_1, \Pi_2} = (f, h, g)$ 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° $h : (I_1, k_1) \in \mathcal{I}_1 \times \mathbb{N} \mapsto [h_{I_1, k_1} \in \mathcal{C}_{I_1}^{C_{I_2}}], \forall x_2 \in \mathcal{C}_{I_2}, h_{I_1, k_1}(x_2) \in \mathcal{C}_{I_1}$ 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 $\rho_2, h \circ (A_k^2) \circ f : (I_1, k_1) \mapsto h_{I_1, k_1}(A_{k_2}^2(I_2))$ est une chaîne garantissant $g(\rho_2)$.

L'expansion représente l'effet de la transformation sur la qualité d'approximation. Une réduction permet alors 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 α_{Π_1, Π_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_2 est indépendante des instances, on dira que α_{Π_1, Π_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 (Paragr. 2.5) 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 définition 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 [42] 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 [44] devient un cas particulier de réduction partielle préservant **APX** (algorithmes à rapport constant) ; l'expansion est alors une homothétie de rapport constant ; en l'appliquant à chaque algorithme d'une chaîne, cette réduction permet de préserver les chaînes à rapport convergeant dans **APX** ;
- la réduction E [35] 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 [17] 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.

2.4. RÉDUCTIONS DU TROISIÈME TYPE

Le cadre que nous venons de décrire reste assez restrictif quant à la forme d'instance à instance (réduction I-I) de la réduction FP. Ce carcan 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 de ne mettre en avant que le 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 I-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 I-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 I-P. Nous analysons son expansion dans la prochaine partie.

2.5. 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, ...

2.5.1. Quelques grands classiques

Le fait d'avoir choisi, pour la réduction FP, le cadre d'instance à instance I-I 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 [43] 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 certains résultats négatifs comme des réductions **FP** transformant un certain niveau d'approximation pour un problème en une résolution exacte pour un autre. Considérons par exemple la proposition 1 de [24]. *Aucun algorithme polynomial ne peut garantir un rapport strictement meilleur que $2/3$ pour BINPACKING sauf si $\mathbf{P} = \mathbf{NP}$.* La preuve met en évidence que si un algorithme polynomial pouvait garantir un rapport strictement plus grand que $2/3$ pour BINPACKING, alors il pourrait résoudre polynomialement le problème de PARTITION, pourtant connu comme étant **NP**-complet.

2.5.2. Stratégie maître-esclave

Revenons à l'exemple du lien entre COLORATION et STABLE que nous avons choisi pour illustrer les réductions I-P. Il s'inscrit dans le schéma général *maître-esclave* décrit dans [31, 44]. 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 I-P du maître à l'esclave. Dans [31], 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 [44] : 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.

2.5.3. Liens entre versions pondérée et non-pondérée

2.5.3.1. 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 [44] 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 1. *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 [44], 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 (*cf.*, par exemple [24]), 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, le rapport $\min\{k/\Delta, k' \log \log \Delta/\Delta\}$ pour **STABLE**, évoqué et discuté dans [24], reste valable pour **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 2. *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 $O(\log^2 n/n \log \log n)$.

2.5.3.2. Cas du stable maximal minimum. Rappelons d'abord un résultat évoqué et discuté dans [24]. Il montre l'impossibilité de résoudre polynomialement

PWSTABLEMAXMIN par des algorithmes à rapports indépendants des poids du graphe-instance.

Proposition 3. *Si $P \neq NP$, alors aucun algorithme ne peut garantir le rapport $n/(4w_{\max})$ pour PWSTABLEMAXMIN.*

Notons maintenant que 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, d'où la proposition suivante.

Proposition 4. *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.*

Cette réduction a dans ce cas un intérêt plus grand. Elle offre la possibilité de récupérer un résultat négatif pour STABLEMAXMIN à partir de la proposition 3. Il en résulte un résultat négatif (Prop. 5) pour le niveau $O(1/n^{1-\epsilon})$. Celui-ci a déjà été obtenu dans [27]; cependant, la preuve résultant de la concaténation des propositions 3, 4 et 5 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** (cf. [24] et annexe B). Il montre aussi que la proposition 3, qui semble être fortement liée aux valeurs des poids, porte en elle également une information d'ordre structurel 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 5. *Si $P \neq NP$, pour tout $\epsilon \in]0, 1[$, STABLEMAXMIN n'est pas approximable au niveau $O(1/n^{1-\epsilon})$.*

Démonstration. 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 4 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 3 montre comment un tel algorithme permettrait de résoudre la 3-colorabilité. \square

2.5.4. 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, 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 I-I. L'idée est exactement la même que celle développée dans la proposition 3 ; 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 différents problèmes.

Proposition 6. *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 $f(kn, \Delta + k - 1)$. Par conséquent, il existe une chaîne d'approximation polynomiale de complexité $O(n^{k/2})$ pour kCl (k fixé) garantissant le rapport d'approximation asymptotique (par rapport à Δ) k/Δ , et une chaîne d'approximation polynomiale de complexité $O(n^k)$ pour STABLE garantissant le rapport d'approximation asymptotique (par rapport à μ) $\min\{k/\Delta, k' \log \log \Delta/\Delta\}$.*

Démonstration. La réduction reprend la construction établie dans la preuve de la proposition 3 dans [24] 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 :

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

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 . \square

Cette réduction préserve en particulier les chaînes à rapport (asymptotique ou non) ρ/Δ . Ceci permet de récupérer, pour kCl , les résultats obtenus pour le stable présentés dans [24], ce qui prouve la deuxième partie de la proposition concernant les rapports d'approximation.

Notons enfin que la proposition 6 permet également de transférer à kCl les résultats d'approximation à rapports fonction de n pour STABLE [14, 23, 24]. Par conséquent, la proposition suivante conclut le paragraphe.

Proposition 7. *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.

2.5.5. 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

complémentaire⁶ \bar{G} , et réciproquement. Chacun sait 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 2.5.3, 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 δ_{\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 1.

- 1° Il existe une réduction I-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 $\rho(\Delta + 1, \Delta)$.

Démonstration. 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. \square

Corollaire 1.

- 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 $O(\log^2 \Delta / \Delta)$.

⁶Les arêtes de G sont des non-arêtes de \bar{G} et réciproquement.

Le point 1 signifie que pour CLIQUE, les paramètres n et Δ sont équivalents. Le point 2, quant à lui, correspond à une amélioration du rapport lorsqu'on passe de STABLE à CLIQUE. 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 8. *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 $\rho(g(\Delta))$.*

Démonstration. 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 résoudre de manière exacte STABLE sur les graphes de degré 3 alors que ce problème est **NP**-complet [26]. \square

2.5.6. 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 que 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 1, 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 [20], 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 2 ([20]). *Il existe une réduction I-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 [20].

3. 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* [24] 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 3.4. Néanmoins, les exemples du paragraphe 3.3 illustrent que ce passage est parfois possible et intéressant.

3.1. 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 [24], 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 : les notions de simplicité introduites respectivement dans [5] et dans [43]. Toutes deux s'expriment en termes d'ordre de difficulté.

La simplicité au sens de [43] 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 [5] 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 [5]. 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 $O(n^k)$. **STABLE** reste simple au sens de [43], par contre **COLORATION** ne l'est pas puisque la k -colorabilité, pour $k \geq 3$ est un problème **NP**-complet [26]. Dans le paragraphe 3.2 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 9.

- 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 [43].
- 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 [38].

Démonstration. 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. \square

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 [43] 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 [43] et de schéma d'approximation différentiel [38] 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 9.

3.2. 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.

3.2.1. 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, dès que π est satisfaite pour un ensemble, elle l'est pour toutes ses parties. 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 d'être stable pour un ensemble de sommets d'un graphe, 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. Par exemple, le résultat suivant est montré dans [28].

Théorème 3. *HG est approximable au niveau $O(\log n/n)$ où n est le cardinal de E .*

3.2.2. 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 et rappelons que chaque instance de Π s'exprime par :

$$\left\{ \begin{array}{l} \text{opt} \quad v(x) \\ x \in \mathcal{C} \\ x_i \in \{0, 1\}, \end{array} \right.$$

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 2. 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 2. Citons notamment H-TRANSVERSAL ou G-TRANSVERSAL, BINPACKING ou encore COLORATION. Par contre, le problème de la coloration minimum d'un graphe 4-colorable⁷ ou encore le problème BINPACKING $_{\mathcal{C}}$ (le problème de BINPACKING restreint aux listes de \mathcal{C} 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 $a + b > 1$) ne sont pas radiaux (la Prop. 11 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 10. *$\sigma(I)$ est un ordre de difficulté pour la résolution exacte pour les problèmes radiaux NP-difficiles.*

3.2.3. 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 11. *Les problèmes radiaux NP-difficiles pour lesquels une distance minimum ϵ sépare deux valeurs réalisables sont simples au sens de [5].*

À l'inverse, le problème de la satisfiabilité pondérée maximum (*max-weighted-sat* [5]) ne l'est pas. Il en est de même pour la coloration minimum d'un graphe 4-colorable ou encore pour le problème BINPACKING $_{\mathcal{C}}$. Ainsi, la proposition 11 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.

⁷Une 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.

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. On peut aussi 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.

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 supérieur 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 STABLE⁸.

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 12. *\tilde{S} est simple au sens de [5] mais non radial.*

Démonstration. 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

⁸Elle correspond à la question suivante : étant donnés G et k , existe-t-il un stable de G de valeur au moins k ?

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}' . \square

3.3. STRUCTURATION PAR ÉLARGISSEMENT D'UNE CLASSE

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

3.3.1. Stabilité d'un algorithme par rapport à une fonction d'éloignement

La notion de *stabilité en approximation* a été introduite dans [10]. 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 peut 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 [10] 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 [10] 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 [10] 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 pour 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. Pour plus d'information sur cette version de TSP, cf. [3, 7, 10–13].

Une manière de généraliser cette notion est de parler de stabilité lorsque Π se réduit à Π' avec une expansion 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 [10] de quasi-stabilité.

3.3.2. 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'arêtes⁹ (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

⁹Où à l'égalité entre la taille d'un couplage maximum et la taille d'un transversal minimum.

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 relaxations 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 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 13. $D(G)$ est un ordre de difficulté pour la résolution exacte.

Démonstration. 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. \square

Remarquons que $D(G) \geq n/2 - \alpha(G)$. Par ailleurs, nous montrons dans la proposition 17 de la section 3.4.3 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 3.4.2, 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 14. *Il existe un algorithme polynomial pour STABLE garantissant le rapport $1/(D(G) + 1)$.*

Démonstration. Considérons l'algorithme suivant :

BEGIN

```

(1) construire un couplage maximum M ;
(2) m ← |M| ;
(3) x ← |X| ;
(4) S ← X ;
(5) FOR k ∈ {0, ..., n - 1} DO
(6)   partitionner M en k + 1 parties M1, ..., Mk+1 de taille ≥ ⌊m/(k + 1)⌋ ;
(7)   M ← {Mi : i = 1...k + 1} ;
(8)   Partitionner X en k + 1 parties X1, ..., Xk+1 de taille ≥ ⌊x/(k + 1)⌋ ;
(9)   X ← {Xi : i = 1...k + 1} ;
(10)  FOR (A, B) ∈ M × X DO
(11)    IF G' = G[X(A) ∪ B] ∈ KE THEN
(12)      S' ← stable maximum de G' ;
(13)    IF |A| < m/(k + 1) AND |B| < x/(k + 1) THEN
(14)      IF ∃u ∈ M \ A tel que G'' = G[X(A ∪ {u}) ∪ B] ∈ KE THEN
(15)        S' ← stable maximum de G'' ;
(16)      ELSE IF ∃v ∈ E \ B tel que G'' = G[X(A) ∪ B ∪ {v}] ∈ KE THEN
(17)        S' ← stable maximum de G'' ;
(18)      FI
(19)    FI
(20)  FI
(21)  S ← 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 [41]. 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, $O(n^{3,5})$ étapes. Cette complexité correspond à la complexité maximale des lignes (14) et (16), la reconnaissance de la propriété KE nécessitant $O(n^{2,5})$ étapes. Il en résulte une complexité globale de l'ordre de $O(n^{6,5})$.

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

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

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 (1))

$$|S'| \geq \frac{m}{D(G)+1} + \frac{|X|}{D(G)+1} - \frac{D(G)}{D(G)+1}. \quad (2)$$

Par ailleurs, $\alpha(G) = m + |X| - D(G)$. La solution construite étant de taille au moins égale à celle de S' , les expressions (2) et celle pour $\alpha(G)$ concluent que le rapport $1/(D(G)+1)$ est garanti. \square

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 13, établit un résultat valable pour tout graphe qui met en évidence comment le rapport d'approximation se détériore en fonction de l'éloignement.

3.4. 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.

3.4.1. *Stable et rapports* $O(n^{\epsilon-1})$

Considérons toujours STABLE pour lequel il est établi [29] 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 $\beta \bar{\theta} f(\sigma)$ 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 se déduit immédiatement de la discussion ci-dessus et met en évidence l’intérêt de la classe $\text{STABLE}_{\geq n^{1-\epsilon}}$.

Proposition 15. *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** car ses instances ne sont pas polynomialement reconnaissables contrairement aux instances d'un problème **NPO**. 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 [36].

3.4.2. 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 3.3.2 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 [2, 14, 22, 33, 36]) 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 ne peuvent être reconnues en temps polynomial.

Il ressort des articles [2, 14] 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 16. *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 .

3.4.3. 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 sa relaxation (noté WSTABLEr) 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.

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

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

Proposition 17 ([39]). *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 solution¹⁰ optimale de WSTABLEr 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 WSTABLEr. Elle correspond, en quelque sorte, aux graphes pour lesquels la solution de la relaxation 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 18. *WSTABLE 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).*

Démonstration. 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. \square

¹⁰La preuve originelle de [39] peut être adaptée pour garantir l'unicité de cette solution.

3.5. 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 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 3 ([38]). Étant donné un problème Π (d'ensembles d'instances \mathcal{I}), un algorithme approché A pour Π et une mesure d'approximation ρ ($\rho \in [0, 1]$) est 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 3.4, 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 [38] s'est d'ores et déjà avéré particulièrement fécond. En particulier, sur l'exemple de BINPACKING, nous mettons évidence dans [21] une famille d'instances critiques pour l'algorithme *first-fit-decreasing* et la mesure différentielle. La preuve illustre comment la définition 3 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 $3/4$.

4. CONCLUSION

Nous avons étudié dans [24] la possibilité de structurer **NPO** en couches d'équi-approximabilité. Dans cette article, nous abordons la question de la structure des problèmes NP-difficiles.

Dans la section 2 nous présentons la notion de réduction qui permet de comparer différents problèmes du point de vue de l'approximation indépendamment des résultats connus pour chacun d'eux. Cet outil 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.

Les 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 désormais le produit de réductions. Certains sont même non égalés par une méthode directe. Ceci 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 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 sans l'être pour un autre. L'exemple de **STABLE** et **CLIQUE** montre l'intérêt de ce type de question ; en particulier, 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 3 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 [24] et dans le paragraphe 3 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 [24] et 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 traiter ces deux rapports de front en paramétrant les résultats et les outils par le type de mesure permet immédiatement d'inclure de nouvelles mesures en tant que paramètres. D'ailleurs, la richesse et la complémentarité entre résultats classiques et différentiels ne peut que nous encourager à étudier d'autres mesures.

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.

ANNEXES

Dans les annexes qui suivent, nous reprenons et complétons la liste des problèmes et les définitions de base sur l'approximation polynomiale présentés dans les annexes de l'article [24].

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 ($kC\ell$)

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 ($WkC\ell$) : 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 sommets¹¹ (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 $v \notin V'$, $V' \cup \{v\}$ 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'ensemble¹² (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 $S' \subseteq \mathcal{S}$ couvrant E , *i.e.*, $\cup_{S \in S'} S = E$;

OBJECTIF : minimiser le cardinal de 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 d'un graphe

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 adjacentes;

OBJECTIF : maximiser $|E'|$;

COMMENTAIRE : le problème est polynomial [41].

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)
 - $\lambda(I)/\beta(I) \geq \rho(I)$, si Π est un problème de maximization, ou
 - $\beta(I)/\lambda(I) \geq \rho(I)$, 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 (connu) 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 **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 **PO** la classe des problèmes d'optimisation polynomiaux. Certaines classes contiennent même des problèmes complets pour des réductions appropriées. Nous rappelons qu'étant donné une classe **C** et une réduction **R**, un problème Π est **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** et 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 réduction appropriée pour prouver les différentes complétudes; justement la réduction **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 **A**; elle est non-vidée;

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 **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 **I-P** du type « maître-esclave » (Paragr. 2.5.2) nous semble une piste intéressante pour la preuve d'une telle existence;

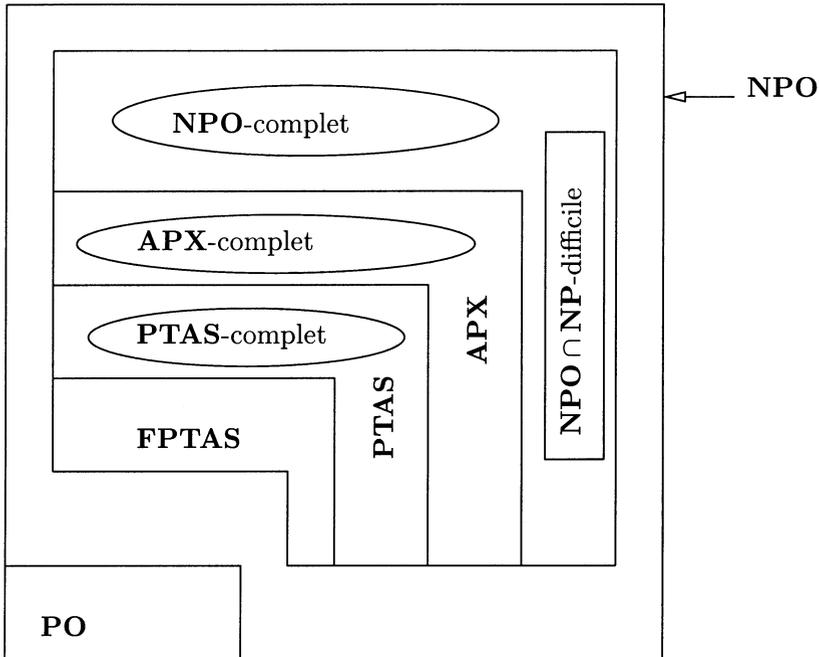


FIGURE 1. Le monde de l’approximabilité selon les conjectures des chercheurs.

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 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 **P** ≠ **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.

Remerciements. Les remarques, commentaires et suggestions des deux lecteurs/lectrices anonymes ont très largement contribué à l'amélioration de la qualité scientifique et de la lisibilité de cet article. Qu'ils/elles en soient remercié(e)s.

RÉFÉRENCES

- [1] L. Alfandari, *Approximation de problèmes de couverture et de partitionnement de graphes*, Ph.D. Thesis. LAMSADE, Université Paris-Dauphine (1999).
- [2] N. Alon et N. Kahale, Approximating the independence number *via* the θ -function. *Math. Programming* (1998).
- [3] T. Andreae et H.-J. Bandelt, Performance guarantees for approximation algorithms depending on parametrized triangle inequalities. *SIAM J. Discrete Math.* **8** (1995) 1-16.
- [4] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela et M. Protasi, *Complexity and approximation. Combinatorial optimization problems and their approximability properties*. Springer, Heidelberg (1999).
- [5] G. Ausiello, P. Crescenzi et M. Protasi, Approximate solutions of NP optimization problems. *Theoret. Comput. Sci.* **150** (1995) 1-55.
- [6] G. Ausiello, A. D'Atri et M. Protasi, Structure preserving reductions among convex optimization problems. *J. Comput. System Sci.* **21** (1980) 136-153.
- [7] M.A. Bender et C. Chekuri, Performance guarantees for the TSP with a parametrized triangle inequality, dans *Proc. WADS'99*. Springer, *Lecture Notes in Comput. Sci.* **1663** (1999) 80-85.
- [8] C. Berge, *Graphs and hypergraphs*. North Holland, Amsterdam (1973).
- [9] P. Berman et J. Hartmanis, On isomorphisms and density of np and other complete sets. *SIAM J. Comput.* **6** (1977) 305-322.
- [10] H.-J. Böckenhauer, J. Hromkovič, R. Klasing, S. Seibert et W. Unger, *Towards the notion of stability of approximation algorithms and the traveling salesman problem*, Report 31, Electr. Colloq. Computational Comp. (1999).
- [11] ———, Approximation algorithms for the TSP with sharpened triangle inequality. *Inform. Process. Lett.* **75** (2000) 133-138.
- [12] ———, An improved lower bound on the approximability of metric TSP and approximation algorithms for the TSP with sharpened triangle inequality, dans *Proc. STACS'00*. Springer, *Lecture Notes in Comput. Sci.* (2000) 382-394.
- [13] H.-J. Böckenhauer et S. Seibert, Improved lower bounds on the approximability of the traveling salesman problem. *RAIRO : Theoret. Informatics Appl.* **34** (2000) 213-255.
- [14] B.B. Boppana et M.M. Halldórsson, Approximating maximum independent sets by excluding subgraphs. *BIT* **32** (1992) 180-196.
- [15] N. Creignou, *Temps linéaire et problèmes NP-complets*, Ph.D. Thesis. Université de Caen (1993).
- [16] P. Crescenzi, A short guide to approximation preserving reductions, dans *Proc. Conference on Computational Complexity* (1997) 262-273.
- [17] P. Crescenzi, V. Kann, R. Silvestri et L. Trevisan, *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/`
- [18] P. CRESCENZI ET A. PANCONESI, Completeness in approximation classes. *SIAM J. Comput.* (1991).
- [19] M. Demange, P. Grisoni et V.T. Paschos, Differential approximation algorithms for some combinatorial optimization problems. *Theoret. Comput. Sci.* **209** (1998) 107-122.
- [20] M. Demange, J. Monnot et V.T. Paschos, Bridging gap between standard and differential polynomial approximation : The case of bin-packing. *Appl. Math. Lett.* **12** (1999) 127-133.
- [21] ———, Maximizing the number of unused bins. *Found. Comput. Decision Sci.* **26** (2001) 169-186.

- [22] M. Demange et V.T. Paschos, Valeurs extrémales d'un problème d'optimisation combinatoire et approximation polynomiale. *Math. Inf. Sci. Humaines* **135** (1996) 51-66.
- [23] ———, *Towards a general formal framework for polynomial approximation*. Cahier du LAMSADE **177**. LAMSADE, Université Paris-Dauphine (2001).
- [24] ———, Autour de nouvelles notions pour l'analyse des algorithmes d'approximation : formalisme unifié et classes d'approximation. *RAIRO : Oper. Res.* **36** (2002) 237-277.
- [25] U. Feige et J. Kilian, Zero knowledge and the chromatic number, dans *Proc. Conference on Computational Complexity* (1996) 278-287.
- [26] M.R. Garey et D.S. Johnson, *Computers and intractability. A guide to the theory of NP-completeness*. W. H. Freeman, San Francisco (1979).
- [27] M.M. Halldórsson, Approximating the minimum maximal independence number. *Inform. Process. Lett.* **46** (1993) 169-172.
- [28] ———, *Approximations via partitioning*, JAIST Research Report IS-RR-95-0003F. Japan Advanced Institute of Science and Technology, Japan (1995).
- [29] J. Hästad, Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Math.* **182** (1999) 105-142.
- [30] D.S. Hochbaum, *Approximation algorithms for NP-hard problems*. PWS, Boston (1997).
- [31] D.S. Johnson, Approximation algorithms for combinatorial problems. *J. Comput. System Sci.* **9** (1974) 256-278.
- [32] V. Kann, Polynomially bounded problems that are hard to approximate. *Nordic J. Comput.* **1** (1994) 317-331.
- [33] D. Karger, R. Motwani et M. Sudan, Approximate graph coloring by semidefinite programming. *J. Assoc. Comput. Mach.* **45** (1998) 246-265.
- [34] R.M. Karp, Reducibility among combinatorial problems, dans *Complexity of computer computations*, édité par R.E. Miller et J.W. Thatcher, Plenum Press, New York (1972) 85-103.
- [35] S. Khanna, R. Motwani, M. Sudan et U. Vazirani, On syntactic versus computational views of approximability. *SIAM J. Comput.* **28** (1998) 164-191.
- [36] J. Lorenzo, *Approximation des solutions et des valeurs des problèmes NP-complets*, Thèse de Doctorat. CERMSEM, Université Paris I (en préparation).
- [37] N. Lynch et J. Lipton, On structure preserving reductions. *SIAM J. Comput.* **7** (1978) 119-126.
- [38] J. Monnot, *Familles critiques d'instances et approximation polynomiale*, Ph.D. Thesis. LAMSADE, Université Paris-Dauphine (1998).
- [39] G.L. Nemhauser, L.A. Wolsey et M.L. Fischer, An analysis of approximations for maximizing submodular set functions. *Math. Programming* **14** (1978) 265-294.
- [40] P. Orponen et H. Mannila, *On approximation preserving reductions : Complete problems and robust measures*, Tech. Rep. C-1987-28. Dept. of Computer Science, University of Helsinki, Finland (1987).
- [41] C.H. Papadimitriou et K. Steiglitz, *Combinatorial optimization : Algorithms and complexity*. Prentice Hall, New Jersey (1981).
- [42] C.H. Papadimitriou et M. Yannakakis, Optimization, approximation and complexity classes. *J. Comput. System Sci.* **43** (1991) 425-440.
- [43] A. Paz et S. Moran, Non deterministic polynomial optimization problems and their approximations. *Theoret. Comput. Sci.* **15** (1981) 251-277.
- [44] H.U. Simon, On approximate solutions for combinatorial optimization problems. *SIAM J. Discrete Math.* **3** (1990) 294-310.
- [45] V. Vazirani, *Approximation algorithms*. Springer, Heidelberg (2001).