



HAL
open science

Structures des classes d'approximation : un état de l'art

Bruno Escoffier, Vangelis Paschos

► **To cite this version:**

Bruno Escoffier, Vangelis Paschos. Structures des classes d'approximation : un état de l'art. 2007.
hal-00948835

HAL Id: hal-00948835

<https://hal.science/hal-00948835v1>

Preprint submitted on 18 Feb 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CAHIER DU LAMSADE

247

Février 2007

Structures des classes d'approximation : un état de
l'art

Bruno Escoffier, Vangelis Th. Paschos

Structures des classes d'approximation : un état de l'art

Bruno Escoffier

Vangelis Th. Paschos

LAMSADE, Université Paris-Dauphine et CNRS UMR 7024
Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16, France
{escoffier,paschos}@lamsade.dauphine.fr

17 février 2007

Résumé

L'objectif de cet article est double : premièrement présenter la problématique de la structuration des classes d'approximation par l'introduction de réductions préservant l'approximabilité de problèmes difficiles d'optimisation et de complétude, et, d'autre part, faire un tour d'horizon sur les principaux résultats obtenus dans ce domaine.

1 Introduction

Très souvent en informatique ou en recherche opérationnelle nous nous trouvons devant la situation suivante : il nous est demandé de résoudre des problèmes d'optimisation (discrète) où le nombre de solutions est potentiellement exponentiel en la taille de l'instance. Un examen exhaustif étant a priori exclu, nous voudrions trouver un algorithme efficace pour les résoudre, mais il semble peu probable qu'il en existe. Que faire?

Une première idée consiste à chercher des algorithmes qui pourraient être très lents (au pire des cas), mais relativement efficaces en pratique. Des méthodes ont ainsi été développées dans ce but, notamment les méthodes de *branch and bound* ou la « programmation dynamique ». On peut également essayer de proposer des algorithmes dont la complexité au pire des cas est certes supra polynomiale mais aussi faible que possible : c'est la problématique de la « complexité au pire des cas » (*worst case complexity*).

Un deuxième axe consiste à privilégier l'aspect efficace de l'algorithme. On va alors essayer de proposer des algorithmes rapides qui, puisqu'ils ne peuvent garantir l'obtention d'une solution optimale, fournissent une solution réalisable aussi bonne que possible. Ce sont souvent des méthodes heuristiques, très satisfaisante dans le cas (souvent avéré) où l'optimalité d'une solution n'est pas absolument requise. Dans cette problématique, il est naturel de s'intéresser à la qualité des solutions fournies par notre algorithme. Dans cet ordre d'idée, on voudrait avoir une garantie quant à cette qualité.

C'est là la problématique de l'approximation polynomiale : construire des algorithmes qui fournissent en un temps polynomial des solutions réalisables ayant une certaine qualité. Ceci implique la définition d'une mesure de la qualité d'une solution : c'est le rôle du rapport d'approximation, que nous aborderons dans la section 3.

La problématique de l'approximation polynomiale est apparue assez tôt, peu de temps après le théorème de COOK initiant le développement de la complexité. En effet, on considère généralement que l'article fondateur du domaine est l'article *Approximation algorithms for combinatorial problem* écrit par JOHNSON en 1974 ([41]). Il y introduit le rapport d'approximation dit standard (défini dans la section 3) et étudie des problèmes classiques pour lesquels il propose des algorithmes (polynomiaux) offrant des garanties de performance. Suite à cet article s'est ensuite développé le domaine de recherche de l'approximation polynomiale. Un très grand nombre de problèmes d'optimisation combinatoire classiques, **NP**-complets dans leur version décision (ce concept est rapidement présenté dans la section 2), ont été étudiés de ce point de vue.

Un des grands enjeux du domaine de l'approximation polynomiale est, dans un premier temps, la structuration de la classe principale des problèmes d'optimisation, appelée **NPO** et présentée dans la section 2, en des sous-classes, les problèmes de chacune d'elles ayant des propriétés communes d'approximabilité, par exemple, ils sont approximables par des rapports d'approximation constants, ou polynomiaux en la taille de l'instance, etc. Les problèmes difficiles de l'optimisation combinatoire (les problèmes **NP**-difficiles, voir section 2), même s'ils peuvent être vus comme équivalents du point de vue de leurs résolutions optimales, se comportent très différemment vis-à-vis de leur résolution par des algorithmes d'approximation et peuvent être classifiés par rapport à ce comportement. Par exemple, bien qu'il y ait des problèmes qui admettent des algorithmes fournissant les solutions dont les valeurs sont très proches de valeurs optimales (par exemple, à une constante indépendante de l'instance près), il y a d'autres problèmes pour lesquels des telles solutions ne peuvent être calculées par aucun algorithme polynomial, à moins qu'une hypothèse de théorie de complexité considérée comme très hautement improbable (par exemple, $\mathbf{P} = \mathbf{NP}$) ne soit pas vraie.

Dans un second temps, l'enjeu devient la structuration de chacune de ces sous-classes en mettant en évidence des problèmes « difficiles » pour chacune d'elles ; cela revient à refaire un travail qui s'apparente en quelque sorte à la **NP**-complétude (dans le cadre de la séparation **P** et **NP**) mais adapté aux caractéristiques combinatoires de chacune des classes d'approximabilité considérée. Cela se fait par l'introduction de notions de réductions préservant l'appartenance dans les classes en question et de complétude pour ces classes. Les sections 4 à 8 sont dédiées à cet enjeu. La technique de la réduction, *i.e.*, de transformation d'un problème en un autre de telle manière qu'une solution pour le deuxième produise une solution pour le premier, est une technique très connue en mathématiques et se trouve au cœur de l'informatique depuis le théorème de COOK en 1971 ([17]). Un aspect très intéressant de ce procédé est le double rôle de la réduction : elle transfère des résultats « positifs » (de possibilité de résolution selon un critère défini *a priori*) du deuxième problème au premier, et des résultats « négatifs » (d'impossibilité de résolution selon le même critère) du premier au

deuxième. Par le biais de telles réductions nous pouvons établir un préordre entre les problèmes d'optimisation en termes de propriétés d'approximabilité et indépendant du niveau actuel¹ de connaissance sur l'approximabilité de chacun de ces problèmes.

Notons enfin que les prémices du concept de réductions préservant l'approximabilité apparaissent dans [41] :

what is it that makes algorithms for different problems behave in the same way? Is there some stronger kind of reducibility than the simple polynomial reducibility that will explain these results, or are they due to some structural similarity between the problems as we define them?

Cet état de l'art reprend et met en perspective les résultats les plus importants concernant cet enjeu de structuration du **NPO** en présentant les principaux résultats obtenus dans ce domaine.

2 Les classes **P**, **NP**, **NPO** et les réductions de **KARP** et de **TURING**

Dans cette section préliminaire nous reprenons quelques notions de base de la théorie de la complexité qui nous serviront plus tard. Pour une introduction plus complètes sur les sujets parcourus ici, le lecteur peut se référer aux ouvrages [36, 56].

La théorie de la complexité est centrée sur les *problèmes de décision*. Ces problèmes sont définis par un ensemble d'instances D , et par une question. Il s'agit alors de déterminer, étant donnée une instance $I \in D$, si la réponse à la question est *oui* ou *non*. Plus formellement, un problème de décision est un ensemble D d'instances partitionné en un ensemble D^+ d'instances positives (celles pour lesquelles la réponse est *oui*) et un ensemble D^- d'instances négatives (celles pour lesquelles la réponse est *non*). Résoudre ce problème consiste à déterminer, étant donnée $I \in D$, si $I \in D^+$ (ou si $I \in D^-$).

La résolution d'un problème se fait grâce à un algorithme. La notion d'algorithme a été formalisée par l'introduction des *machines de TURING*. Intuitivement, une machine de TURING déterministe possède une tête de lecture/écriture sur un ruban divisé en cases numérotées. Dans chaque case est écrit un symbole choisi parmi $\{0,1,b\}$. La taille du ruban est potentiellement infinie. La machine possède également un état, évoluant parmi un ensemble Γ . La machine fonctionne alors de la manière suivante : elle commence à lire la case 1, et son état initial est un état $e \in \Gamma$ particulier. A chaque fois qu'elle lit un symbole, et en fonction de son état, elle inscrit un symbole dans la case courante (en effaçant le symbole qui y était inscrit), elle se déplace d'une case vers la droite ou vers la gauche, elle change éventuellement d'état. Le fonctionnement de la machine peut donc être décrit par une fonction de $\{0,1,b\} \times \Gamma$ dans $\{0,1,b\} \times \{-1, +1\} \times \Gamma$.

Une instance I d'un problème (codée en binaire) est écrite dans les cases 1 à $|I|$ (les autres cases comportant le symbole b). La machine effectue son calcul (selon le fonc-

1. Niveau qui peut changer du jour au lendemain par la découverte d'un meilleur algorithme ou d'un résultat plus fort d'approximabilité.

tionnement précédent) et s'arrête lorsqu'elle est dans un des deux états f_{oui} et f_{non} , appelés *états terminaux*. Le temps d'exécution est mesuré par le nombre d'étapes effectuées. L'instance I est acceptée par la machine si le calcul se termine dans l'état f_{oui} , il est rejeté sinon. Le langage reconnu par la machine est défini comme étant l'ensemble des instances acceptées par cette machine. Nous dirons alors qu'une machine résout le problème de décision $D = (D^+, D^-)$ si le langage reconnu par la machine est D^+ .

L'aspect considéré le plus important pour la résolution d'un problème est sa *complexité en temps*, c'est-à-dire le nombre d'étapes que va effectuer la machine pour déterminer si $I \in D^+$. Il est généralement considéré qu'un algorithme est efficace quand cette complexité en temps est bornée par une fonction polynomiale en la taille de l'instance proposée à la résolution. Cela motive l'introduction de la classe **P**. Formellement, la classe **P** est l'ensemble des problèmes de décision pouvant être résolus en temps polynomial par une machine de TURING déterministe.

Cependant, pour un grand nombre de problèmes de décision naturels, nous ne connaissons pas d'algorithme polynomial permettant de les résoudre. Le problème MAX STABLE en est un exemple. Dans sa version décisionnelle ce problème consiste à déterminer si un graphe G contient un ensemble stable (c'est-à-dire un sous-ensemble de sommets deux à deux non adjacents) de taille supérieure ou égale à un entier K . Néanmoins, pour ce problème, si nous ne sommes pas capables (à l'heure actuelle) de déterminer efficacement (*i.e.*, en temps polynomial) si une instance (G, K) est positive, nous pouvons, étant donné un ensemble de sommets, déterminer aisément si c'est un stable et si son cardinal est supérieur ou égal à K . Autrement dit, nous pouvons vérifier qu'un ensemble de sommets constitue une preuve certifiant le fait que l'instance est positive. Cette capacité de tester une preuve est un trait commun à un grand nombre de problèmes. Ceci motive la définition de la classe **NP**. Formellement, un problème de décision appartient à **NP** s'il existe une relation binaire polynomiale R et un polynôme p tels que : pour toute instance $I \in D$, $I \in D^+$ si et seulement si il existe x tel que $R(I, x)$ et $|x| \leq p(|I|)$. Par relation binaire polynomiale, nous entendons qu'étant donnés I et x , on peut déterminer en temps polynomial en fonction de $|I| + |x|$ si $R(I, x)$ ou non. On dit que x est un *certificat* prouvant que l'instance I est positive. On peut caractériser les problèmes de cette classe comme ceux résolubles en temps polynomial par une machine de TURING d'un type plus général que celles précédemment définies, appelée machine de TURING non déterministe : un problème de décision appartient à **NP** s'il est résoluble par une machine de TURING non déterministe polynomiale. Très brièvement, dans une machine non déterministe, la fonction régissant le fonctionnement de cette machine est multivaluée. La machine a plusieurs possibilités à chaque itération, et une instance est acceptée s'il existe au moins un ensemble de choix conduisant à l'état d'acceptation f_{oui} .

La question centrale consiste bien évidemment à savoir si $\mathbf{P} = \mathbf{NP}$ (l'inclusion de **P** dans **NP** étant évidente). L'importance majeure de cette question, formellement posée depuis plus de trente ans, n'est plus à démontrer. L'intérêt de la plus grande partie du domaine de la théorie actuelle de la complexité n'est véritablement manifeste que sous l'hypothèse, généralement admise, que $\mathbf{P} \neq \mathbf{NP}$.

Devant l'apparente difficulté de répondre à la question $\mathbf{P} = \mathbf{NP}$?, l'idée à alors été

lancée d'étudier les relations entre problèmes. *Peut-on dire que tel problème est plus difficile à résoudre que tel autre? Peut-on déduire pour un problème quelque chose du caractère facile ou difficile de la résolution d'autres problèmes? Peut-on obtenir une notion de hiérarchie de difficulté relative à la résolution des problèmes?* La notion de réduction de KARP a été introduite pour répondre à ces interrogations.

Soient $D_1 = (D_1^+, D_1^-)$ et $D_2 = (D_2^+, D_2^-)$ deux problèmes de décision. On dit que D_1 se réduit à D_2 sous une réduction de KARP (ce que nous noterons $D_1 \leq_K D_2$) s'il existe une fonction f telle que :

- $\forall I \in D_1, f(I) \in D_2$;
- $I \in D_1^+ \Leftrightarrow f(I) \in D_2^+$;
- f est calculable en temps polynomial.

La propriété fondamentale de cette réduction est que si D_2 est dans \mathbf{P} et si $D_1 \leq_K D_2$, alors D_1 est dans \mathbf{P} . En effet, pour obtenir un algorithme polynomial résolvant D_1 , il suffit, étant donnée une instance I de D_1 , de calculer $f(I)$ et d'utiliser l'algorithme polynomial résolvant D_2 (notons que puisque f est calculable en temps polynomial, $|f(I)|$ est polynomialement borné en fonction de $|I|$). En d'autres termes, *la réduction de KARP préserve l'appartenance à \mathbf{P} , c'est-à-dire, que si $D_2 \in \mathbf{P}$ et $D_1 \leq_K D_2$, alors $D_1 \in \mathbf{P}$.*

Une réduction de KARP d'un problème D_1 à un problème D_2 indique donc que le problème D_2 est plus difficile à résoudre que le problème D_1 , en ce sens que si l'on savait résoudre D_2 en temps polynomial, on saurait également résoudre D_1 en temps polynomial. Voulant déterminer si un problème donné est dans \mathbf{P} , plus il y a de problèmes (dont le statut est indéterminé, polynomiaux ou non) qui se réduisent à lui, moins il semble probable qu'il soit polynomial. Une interrogation naturelle consiste alors à savoir s'il existe un problème de \mathbf{NP} plus difficile que tous les autres, c'est-à-dire tel que tout problème de \mathbf{NP} se réduit (sous une réduction de KARP) à lui. De tels problèmes seront dits complets pour la classe \mathbf{NP} (sous la réduction de KARP), ou plus simplement \mathbf{NP} -complets (la référence à la réduction de KARP sera généralement omise). Donc, un problème $D \in \mathbf{NP}$ est dit \mathbf{NP} -complet² (sous la réduction de KARP) si tout problème de \mathbf{NP} se réduit à D (sous cette réduction). Un corollaire de cela est qu'un problème D \mathbf{NP} -complet est polynomial si et seulement si $\mathbf{P} = \mathbf{NP}$. Cette remarque stipule donc que la classe \mathbf{P} et celle des problèmes \mathbf{NP} -complets sont disjointes (si $\mathbf{P} \neq \mathbf{NP}$). Voyant que les problèmes usuels sont soit polynomiaux soit \mathbf{NP} -complets, on peut alors légitimement se demander si ces deux classes forment une partition de \mathbf{NP} . LADNER a répondu par la négative ([47]). *Il existe des problèmes qui ne sont ni \mathbf{NP} -complets, ni dans \mathbf{P} si $\mathbf{P} \neq \mathbf{NP}$.* De tels problèmes sont appelés problèmes \mathbf{NP} -intermédiaires (par rapport à \mathbf{P} , sous la réduction de KARP). LADNER prouve même qu'il existe une infinité de niveaux dans la hiérarchie créée par la réduction de KARP.

2. Le premier problème démontré \mathbf{NP} -complet (par COOK, [17]) est le problème SAT dont une instance est constituée de variables booléennes et d'un ensemble de clauses (disjonctions de littéraux) sur ces variables ; il s'agit alors de déterminer s'il existe une valeur de vérité des variables booléennes satisfaisant toutes les clauses données en entrée.

La construction de ces problèmes intermédiaires soulève une question concernant la puissance de la réduction de KARP. Dans cette réduction, nous transformons une instance du problème de départ en une instance du problème d'arrivée de manière à pouvoir résoudre l'instance du problème de départ si l'on sait résoudre l'instance du problème d'arrivée. Nous l'avons indiqué, la propriété fondamentale de cette réduction est qu'elle préserve l'appartenance à \mathbf{P} . Cependant, nous pouvons remarquer que si au lieu de ne construire (et de ne résoudre) qu'une instance du problème d'arrivée, nous en construisons plusieurs (un nombre raisonnable bien sûr), nous conservons la propriété de la préservation de l'appartenance à \mathbf{P} . La restriction de la réduction de KARP (une seule instance du problème d'arrivée) ne semble pas capitale si notre but est de définir une notion de \mathbf{NP} -complétude. Cette remarque motive la définition de la *réduction de TURING*. Soient D et D' deux problèmes de décision. On dit que D se réduit à D' sous la réduction de TURING (ce que l'on notera $D \leq_T D'$) si, étant donné un oracle \bigcirc résolvant le problème D' , il existe un algorithme polynomial A résolvant D en faisant appel à \bigcirc .

Précisons le sens de cette définition. Un oracle pour le problème D' est une sorte d'algorithme imaginaire qui, sur une instance I de D' , résout le problème, c'est-à-dire détermine si $I \in D'^+$. Nous considérerons que son temps d'exécution est unitaire. Cette définition traduit donc effectivement l'idée selon laquelle on peut résoudre un nombre polynomial d'instances du problème d'arrivée.

Soient D et D' deux problèmes de \mathbf{NP} . Si D' appartient à \mathbf{P} et si $D \leq_T D'$, alors $D \in \mathbf{P}$. Autrement dit, *la réduction de TURING préserve l'appartenance à \mathbf{P}* . En effet, il suffit d'utiliser l'algorithme A où, à la place de l'appel à l'oracle, nous exécutons l'algorithme supposé polynomial résolvant D' .

Nous pouvons alors définir la notion de \mathbf{NP} -complétude sous la réduction de TURING. *Un problème de \mathbf{NP} est dit \mathbf{NP} -complet sous la réduction de TURING si tout problème de \mathbf{NP} se réduit à lui sous la réduction de TURING.*

Nous pouvons également définir de manière analogue les problèmes \mathbf{NP} -intermédiaires (relativement à \mathbf{P}) sous la réduction de TURING. Nous devons donc maintenant étudier la puissance de cette nouvelle réduction. Est-elle plus puissante que la réduction de KARP? Y a-t-il des problèmes intermédiaires sous cette réduction? En réalité, le pouvoir de cette réduction paraît être encore mal cerné. Il semble cependant qu'elle soit strictement plus large que la réduction de KARP ([48]). La question de l'existence de problèmes intermédiaires reste à notre connaissance ouverte.

Ce qui est très intéressant dans la réduction de TURING et qui sera très explicitement utilisé dans les réductions préservant l'approximabilité est l'idée de faire appel à un algorithme pour un problème pour résoudre un autre problème. Elle s'adapte donc très bien à des problèmes quelconques (d'optimisation par exemple). En effet, on peut dire qu'un problème se réduit à un autre problème si l'on peut résoudre le premier en temps polynomial à l'aide d'un algorithme utilisant un oracle pour résoudre le second. Suivant cette remarque, nous pouvons alors vouloir déterminer si un problème (pas forcément dans \mathbf{NP}) est plus difficile à résoudre que tous les problèmes de \mathbf{NP} . Donc, *un problème Π est dit \mathbf{NP} -difficile sous la réduction de TURING si tout problème de \mathbf{NP} se réduit à lui sous une telle réduction.*

Dans ce qui précède, nous avons introduit les problèmes à la base de la théorie de la complexité, les problèmes de décision. On peut cependant remarquer que bon nombre de ces problèmes sont issus de problèmes où l'on doit trouver une solution optimisant un certain critère, problèmes dits *d'optimisation*. Les problèmes qui vont nous intéresser plus particulièrement, sont les problèmes combinatoires. *Grosso modo*, ce sont des problèmes d'optimisation où le nombre de solutions réalisables est fini mais très grand (non polynomialement borné généralement), et où l'on doit trouver la meilleure. Ces problèmes se rapprochent d'une certaine manière des problèmes de décision de **NP**; cela a motivé l'introduction d'une classe particulière de problèmes d'optimisation, la classe **NPO**.

Définition 1. Un problème $\Pi = (\mathcal{I}, \text{sol}, m, \text{but})$ de **NPO** est un quadruplet vérifiant les conditions suivantes :

- \mathcal{I} est l'ensemble des instances ; on peut reconnaître en temps polynomial si une entrée I est une instance du problème ;
- étant données $I \in \mathcal{I}$ et $x \in \text{sol}(I)$, $m(I, x)$ désigne la valeur de la solution x de l'instance I ; il existe un polynôme p tel que pour tout $I \in \mathcal{I}$ et tout $x \in \text{sol}(I)$, $|x| \leq p(|I|)$; de plus, on peut reconnaître en temps polynomial si un mot x est une solution réalisable ou non ;
- on peut trouver en temps polynomial une solution réalisable à toute instance $I \in \mathcal{I}$;
- étant données $I \in \mathcal{I}$ et $x \in \text{sol}(I)$, $m(I, x)$ désigne la valeur de la solution x de l'instance I ; $m(I, x) \in \mathbb{N}^*$, et m est calculable en temps polynomial.
- $\text{but} \in \{\min, \max\}$ indique si l'on veut maximiser ou minimiser.

Résoudre le problème sur l'instance I consiste à déterminer une solution :

$$x^* \in \text{argbut}\{m(I, x), x \in \text{sol}(I)\} \quad \blacksquare$$

On peut dériver un problème de décision d'un problème d'optimisation, en se donnant une instance I et un entier K . Il s'agit alors de déterminer s'il existe une solution de valeur au moins (resp., au plus) K pour un problème de maximisation (resp., de minimisation).

Etant donnée une classe $\mathbf{C} \subseteq \mathbf{NPO}$, nous notons **Max-C** et **Min-C** les restrictions de **C** aux problèmes de maximisation et de minimisation, respectivement.

Soit $\Pi \in \mathbf{NPO}$. On dit que Π est *polynomialement borné* s'il existe un polynôme p tel que pour toute instance I de Π et toute solution réalisable x de I , $m(I, x) \leq p(|I|)$. On dit que Π est *polynomialement borné au sens différentiel* s'il existe un polynôme p tel que pour toute instance I et pour tout couple de solutions $(x, y) \in \text{sol}(I) \times \text{sol}(I)$, $|m(I, x) - m(I, y)| \leq p(|I|)$. Etant donnée une classe $\mathbf{C} \subseteq \mathbf{NPO}$, nous notons **C-PB** (resp., **C-DPB**) la restriction de **C** aux problèmes polynomialement bornés (resp., polynomialement bornés au sens différentiel).

3 Approximation polynomiale

3.1 Rapports d'approximation

La problématique de l'approximation polynomiale peut se résumer en deux mots en *la résolution d'un problème difficile par des algorithmes polynomiaux fournissant toujours des solutions réalisables ayant une certaine qualité*. Ceci implique la définition d'une mesure de la qualité d'une solution : c'est le rôle du rapport d'approximation.

Soit x une solution réalisable d'une instance I d'un problème Π de **NPO**. Le *rapport d'approximation standard* de x sur I est : $\rho_{\Pi}(I,x) = m(I,x)/\text{opt}(I)$. Ce rapport est donc dans $[0,1]$ si $\text{but}(\Pi) = \max$ et dans $[1,\infty[$, $\text{but}(\Pi) = \min$. Dans les deux cas, plus il est proche de 1, meilleure est la solution. Notons que la référence au problème Π sera souvent omise lorsqu'il n'y a pas d'ambiguïté. C'est en mesurant la qualité d'une solution à l'aide de ce rapport qu'a été menée l'immense majorité des travaux sur l'approximation.

Plus d'informations sur la problématique, les outils et méthodes utilisés, et les principaux résultats obtenus dans le domaine de l'approximation polynomiale, le lecteur peut trouver dans les ouvrages anglophones [7, 63] et dans l'ouvrage francophone [56].

Néanmoins, si le succès du rapport d'approximation standard est indéniable, il n'est pas le seul à être utilisé. S'il paraît tout à fait naturel, nous pouvons émettre quelques réserves. Dans un premier temps, il peut sembler intéressant de ne pas jauger la qualité d'une solution au regard uniquement de la valeur optimale. En effet, on peut aussi analyser comment se situe notre solution vis-à-vis de l'ensemble des solutions réalisables, ou vis-à-vis de l'ensemble des valeurs réalisables. Deuxièmement, considérons un problème Π de **NPO**, et définissons un problème Π' où les instances, les solutions réalisables et le but sont les mêmes que pour Π , seule change la valeur d'une solution : $m_{\Pi'}(I,x) = \alpha m_{\Pi}(I,x) + \beta$, où $\alpha > 0$ et β sont deux constantes. Les deux problèmes Π et Π' sont en quelque sorte équivalents. On peut en effet considérer que leur résolution est identique, tout algorithme pour l'un fournit un algorithme pour l'autre dont on aimerait dire qu'il a la même efficacité que le premier. Autrement dit, il paraît souhaitable d'avoir une mesure qui soit stable lorsque l'on opère une simple transformation affine de la valeur des solutions. Ces deux remarques ont conduit AUSIELLO *et al.* ([9]) à définir le *rapport différentiel*. Dans ce rapport, on mesure la qualité d'une solution non plus seulement en fonction de la valeur d'une solution optimale, mais également en fonction de la valeur d'une pire solution. Formellement, cette solution est définie comme une solution optimale d'un problème défini sur le même ensemble d'instances et avec le même ensemble de solutions réalisables mais avec but l'inverse à celui de Π . Cette valeur (notée $\omega(I)$) est correctement définie pour les problèmes de **NPO** (on a pour chaque instance un ensemble fini de solutions réalisables). En d'autres termes, $\omega(I)$ est la valeur d'une solution optimale du problème obtenu en changeant simplement le but du problème (maximiser au lieu de minimiser, ou *vice versa*). Notons que cette valeur peut être facile ou difficile à calculer, selon le problème en question.

Soit x une solution réalisable d'une instance I d'un problème Π de **NPO**. Le

rapport d'approximation différentiel de x sur I est :

$$\delta_{\Pi}(I, x) = \frac{m(I, x) - \omega(I)}{\text{opt}(I) - \omega(I)}$$

Par convention, lorsque $\text{opt}(I) = \omega(I)$, on fixera $\delta_{\Pi}(I, x) = 1$ pour toute solution x (puisque toute solution est optimale). Aussi, la dissymétrie entre maximisation et minimisation s'efface ici puisque dans les deux cas le rapport différentiel est dans $[0, 1]$ (et plus il est proche de 1, meilleure est la solution).

Bien qu'introduit en 1977 et donc peu de temps après l'article de JOHNSON ([41]), ce rapport a été très ponctuellement utilisé jusqu'au début des années 90. En effet, [10, 1, 64] sont les travaux que nous connaissons jusqu'en 1993 quand les prémices d'une approche axiomatique pour l'approximation polynomiale qui a conduit au rapport différentiel fut présentée dans [26] ; cette approche fut complétée dans [27].

La relation suivante lie les rapports standard et différentiel pour un problème de maximisation. Notons qu'aucune relation aussi directe n'existe pour les problèmes de minimisation.

Proposition 1. *Si Π est un problème de maximisation, alors pour toute instance I de Π et toute solution x de I , $\delta_{\Pi}(I, x) \leq \rho_{\Pi}(I, x)$. Par conséquent, si un problème de maximisation est δ -approximable selon la mesure différentielle, alors il est δ -approximable selon la mesure standard.*

Avec la définition de rapports d'approximation, nous avons maintenant une mesure de la qualité d'une solution. Nous pouvons alors introduire précisément la notion basique d'algorithmes approchés.

Soient Π un problème d'optimisation, et ρ une fonction $\rho : \mathcal{I} \rightarrow \mathbb{R}^+$. Un *algorithme approché* A pour Π est un algorithme fournissant, pour toute instance I de Π , une solution réalisable x pour I . On dit que A est ρ -approché si, pour toute instance I , le rapport le rapport d'approximation de x est meilleur que $\rho(I)$. Soient un problème d'optimisation Π et une fonction $\rho : \mathcal{I} \rightarrow \mathbb{R}^+$; Π est dit *approximable à rapport ρ* (ou ρ -approximable) s'il existe un algorithme polynomial ρ -approché pour Π .

Ces définitions sont valables pour un problème de maximisation ou de minimisation, pour l'approximation standard ou pour l'approximation différentielle. Notons que nous nous intéresserons bien sûr à des algorithmes polynomiaux (ce qui sera souvent sous-entendu par la suite) et principalement à des garanties (constantes ou) dépendant uniquement de la taille de l'instance. Dans ce cas, étant donnée une fonction ρ de \mathbb{N} dans \mathbb{R}^+ , on dira bien évidemment que l'algorithme A est ρ -approché si pour toute instance I le rapport d'approximation de A est meilleur que $\rho(|I|)$.

La problématique de l'approximation polynomiale consiste alors, étant donné un problème de **NPO**, à déterminer des garanties de performance les meilleures possibles pour ce problème. Cette recherche revêt deux aspects. Il s'agit dans un premier temps de proposer des algorithmes approchés (et de les analyser) afin d'obtenir des garanties de performance. On parle alors de résultats *positifs* d'approximation. Une première question naturelle consiste à ce stade à déterminer si l'analyse de notre algorithme est

optimale, c'est-à-dire à savoir s'il existe des instances pour lesquelles le rapport proposé est atteint, ou si la garantie fournie par l'algorithme est meilleure mais notre analyse n'a pas pu la révéler. Cela concerne un algorithme particulier ; néanmoins, on doit se poser la question plus générale et beaucoup plus délicate de savoir non pas si notre analyse est bonne, mais si notre algorithme est bon. Existe-t-il d'autres algorithmes meilleurs (au sens des rapports d'approximation) ? Cela conduit tout naturellement à la problématique de la recherche de résultats *négatifs*. Il s'agit alors de prouver que notre problème n'est pas approximable à tel rapport d'approximation, sous une hypothèse de complexité. Autrement dit, il s'agit de prouver que si l'on disposait d'un algorithme garantissant un certain rapport d'approximation, alors on aurait (par exemple) $\mathbf{P} = \mathbf{NP}$. Cette problématique fait l'objet de la section 3.3. On peut alors résumer l'étude d'un problème dans le cadre de l'approximation polynomiale comme suit. Il s'agit à la fois : d'établir que le problème est approximable à rapport ρ (pour le meilleur ρ possible), d'établir que sous une hypothèse de complexité (idéalement $\mathbf{P} \neq \mathbf{NP}$) ce problème n'est pas approximable à rapport ρ' et de chercher à minimiser la différence entre ρ et ρ' . La situation idéale est bien entendu atteinte quand les deux bornes (approximabilité et inapproximabilité) sont arbitrairement proches. Dans un tel cas, nous pouvons considérer que l'on a réussi à déterminer exactement ce qu'il était possible (et ce qu'il n'était pas possible) de faire, en termes d'approximation polynomiale cela s'entend. Ce type de résultats a été obtenu pour certains problèmes, par exemple le très fameux problème MAX E3SAT . Une instance de ce problème consiste en la donnée de clauses sur un ensemble de variables booléennes, chaque clause comportant exactement trois littéraux. Il s'agit alors de trouver une valeur de vérité maximisant le nombre de clauses satisfaites. JOHNSON a montré dans [41] que ce problème est approximable à rapport (standard) $7/8$, tandis que HÅSTAD a montré plus récemment ([39]) qu'il n'est pas approximable à rapport (standard) $7/8 + \varepsilon$, quel que soit $\varepsilon > 0$, sauf si $\mathbf{P} = \mathbf{NP}$.

3.2 Classes d'approximation

Il est très vite apparu, dans l'étude de l'approximation polynomiale des problèmes de \mathbf{NPO} , que ces problèmes avaient des comportements très différents en termes d'approximabilité. Si pour certains problèmes on obtenait des algorithmes à rapport constant par exemple (comme MAX E3SAT), pour d'autres la tâche semblait plus ardue (comme le problème MAX STABLE). Pour différencier les problèmes selon leurs propriétés d'approximabilité, vient alors l'idée de les regrouper dans des classes reflétant un niveau d'approximabilité. Ont ainsi émergé des classes naturelles regroupant ces problèmes.

Un premier ensemble est constitué des problèmes approximables à rapport constant (c'est-à-dire indépendant de l'instance). Un problème Π de \mathbf{NPO} est dans \mathbf{APX} (resp., \mathbf{DAPX}) s'il existe une constante ρ telle que Π est approximable à rapport standard (resp., différentiel) ρ .

L'appartenance d'un problème à \mathbf{APX} montre que son comportement vis-à-vis de l'approximation est plutôt satisfaisant. Néanmoins, pour certains problèmes, on peut faire beaucoup mieux : on peut se fixer n'importe quelle marge de tolérance, une garan-

tie requise arbitrairement bonne, et obtenir un algorithme fournissant cette garantie de performance. C'est la classe des problèmes admettant un schéma d'approximation. Un problème Π de **NPO** est dans **PTAS** (resp., **DPTAS**) si pour toute constance $\varepsilon > 0$, Π est approximable à rapport standard $1 + \varepsilon$ si $\text{but}(\Pi) = \min$, à rapport standard $1 - \varepsilon$ si $\text{but}(\Pi) = \max$ (resp., à rapport différentiel $1 - \varepsilon$). Une famille $(A_\varepsilon)_{\varepsilon>0}$ d'algorithmes $1 - \varepsilon$ (ou $1 + \varepsilon$ suivant les cas) approchés est appelée schéma d'approximation (standard ou différentiel) pour Π . Par ailleurs, un problème Π de **NPO** est dans **FPTAS** (resp., **DFPTAS**) s'il existe un schéma d'approximation standard (resp., différentiel) $(A_\varepsilon)_{\varepsilon>0}$ tel que la complexité de A_ε est polynomiale en la taille de l'instance et en $1/\varepsilon$. Ceci représente une situation quasiment idéale puisque nous pouvons approcher un problème avec une garantie de performance arbitrairement bonne, sans que la complexité n'« explose ».

Nous nous sommes intéressés aux problèmes approximables à rapport constant. Or, comme mentionné en introduction de cette section, il semble exister des problèmes pour lesquels on ne peut trouver d'algorithme approché à rapport constant. On peut dans ce cas chercher à approcher le problème par une fonction ρ de la taille de l'instance. Cette fonction tendra donc vers l'infini (dans le cas d'une minimisation en approximation standard) ou vers 0 (dans les autres cas). Il s'agit alors de trouver la « meilleure » fonction ρ , c'est-à-dire, celle qui tend le moins vite vers l'infini ou vers 0. Les deux classes de fonctions les plus classiques sont les fonctions polynomiales et logarithmiques. Un problème Π de **NPO** est dans **Poly-APX** (resp., **Poly-DAPX**) s'il existe un polynôme p (à valeurs strictement positives) tel que Π est approximable à rapport standard p si $\text{but}(\Pi) = \min$, $1/p$ si $\text{but}(\Pi) = \max$ (resp., à rapport différentiel $1/p$). Par ailleurs, un problème Π de **NPO** est dans **Log-APX** (resp., **Log-DAPX**) s'il existe une constante c telle que Π est approximable à rapport standard $c \log(\cdot)$ si $\text{but}(\Pi) = \min$, $1/(c \ln(\cdot))$ si $\text{but}(\Pi) = \max$.

Ces classes reflètent une hiérarchie dans la difficulté d'approcher des problèmes. Nous avons les inclusions évidentes suivantes :

$$\mathbf{PO} \subset \mathbf{FPTAS} \subset \mathbf{PTAS} \subset \mathbf{APX} \subset \mathbf{Log-APX} \subset \mathbf{Poly-APX} \subset \mathbf{NPO}$$

De plus, toutes ces inclusions sont strictes si $\mathbf{P} \neq \mathbf{NP}$. Nous connaissons même pour chaque inclusion des problèmes naturels prouvant que l'inclusion est stricte. Par exemple, $\text{SAC-A-DOS} \in \mathbf{FPTAS} \setminus \mathbf{PO}$, $\text{MAX STABLE PLANAIRE} \in \mathbf{PTAS} \setminus \mathbf{FPTAS}$, $\text{MAX E3SAT} \in \mathbf{APX} \setminus \mathbf{PTAS}$, $\text{MIN SET COVER} \in \mathbf{Log-APX} \setminus \mathbf{APX}$, $\text{MAX STABLE} \in \mathbf{Poly-APX} \setminus \mathbf{Log-APX}$.

La situation est parfaitement analogue dans le cas de l'approximation différentielle (la hiérarchie est la même et les inclusions sont strictes), à ceci près que l'on ne connaît pas à l'heure actuelle de problème naturel dans la classe **Log-DAPX**.

Mentionnons également que dans le cadre de l'approximation différentielle, une autre classe a été définie : la classe **0-DAPX** ([15]). Il s'agit de la pire situation, à savoir quand pour un problème n'importe quel algorithme (polynomial) fournit la pire solution sur au moins une instance ou (c'est équivalent) sur une infinité d'instances. Autrement dit, ce sont les problèmes pour lesquels on ne peut garantir aucun rapport

s'exprimant comme une fonction ne s'annulant pas. Un problème Π de **NPO** est dans **0-DAPX**, si pour toute fonction $\delta : \mathbb{N} \rightarrow \mathbb{R}_*^+$, Π n'est pas δ -approximable si $\mathbf{P} \neq \mathbf{NP}$. Cette classe est bien entendu d'intersection vide avec la classe **Poly-DAPX** (ainsi qu'avec les classes **Log-DAPX**, **DAPX**, ...). Elle contient cependant des problèmes classiques, par exemple le problème **MIN INDEPENDENT DOMINATING SET** consistant à trouver le plus petit sous-ensemble de sommets qui soit à la fois dominant et stable ([15]).

Dès le début, des recherches ont été menées pour essayer de déterminer pourquoi certains problèmes étaient beaucoup mieux approximables que d'autres. Peut-on trouver des raisons structurelles, peut-on donner des conditions nécessaires ou suffisantes (voire des caractérisations) quant à l'existence d'algorithmes à rapports constants?, de schémas?, de schémas complets? Le *survey* d'Ausiello *et al.* [8] présente un ensemble de résultats s'inscrivant dans cette problématique.

Mentionnons dans cet ordre d'idée la propriété classique suivante : un problème Π de **NPO-PB** (resp., **NPO-DPB**), ne peut admettre de schéma complet standard (resp., différentiel) sauf s'il est polynomial. La preuve est immédiate : si l'on a un schéma complet A_ε et un polynôme p bornant la valeur des solutions réalisables, alors l'algorithme $A_{1/(p(|I|+1))}$ est polynomial et exact.

3.3 Inapproximabilité

L'analyse d'un problème combinatoire dans le cadre de l'approximation polynomiale comporte, nous l'avons déjà mentionné, deux aspects complémentaires : le développement d'algorithmes approchés, et la recherche de résultats négatifs. C'est ce deuxième aspect qui fait l'objet de cette section. Nous introduisons tout d'abord la notion de réduction **GAP**, ayant conduit aux premiers résultats d'inapproximabilité, puis la technique, plus générale, utilisant des réductions préservant l'approximation. Nous abordons brièvement dans un dernier temps les systèmes de preuves probabilistes, sur lesquels se basent la majeure partie des meilleurs résultats négatifs actuels.

3.3.1 Technique initiale : la réduction **GAP**

Un résultat d'inapproximabilité pour un problème Π de **NPO** consiste à prouver que si l'on disposait d'un algorithme ρ -approché (pour un certain ρ), alors on contredirait une hypothèse de complexité, par exemple $\mathbf{P} \neq \mathbf{NP}$. L'idée qui vient alors immédiatement en tête consiste à adapter la réduction de **KARP** pour atteindre l'objectif fixé.

Exemple 1. Reprenons la démonstration de la **NP**-complétude du problème de décision **COLORATION** (dérivé du problème d'optimisation **MIN COLORATION**) proposée dans [36]. Dans le problème **MIN COLORATION**, une instance est un graphe $G(V, E)$. Le but est, rappelons le, de trouver une partition de cardinal minimal des sommets en stables. La réduction proposée consiste à partir du problème **E3SAT** (restriction du problème de décision **SAT** aux instances où les clauses sont de taille exactement 3).

Etant donnée une instance φ de E3SAT , on construit un graphe G tel que si φ est satisfiable, alors G est 3-coloriable, sinon G est au mieux 4-coloriable.

Supposons maintenant que nous ayons un algorithme $(4/3 - \varepsilon)$ -approché (avec $\varepsilon > 0$) pour MIN COLORATION . Partons d'une instance φ de E3SAT , et appliquons notre algorithme sur le graphe G obtenu dans la réduction. Si φ est négative, alors bien évidemment nous obtenons une coloration de taille au moins 4. En revanche, si φ est positive, G est 3-coloriable donc l'algorithme fournit une coloration de taille au plus $3 \times (4/3 - \varepsilon) < 4$, donc une coloration de taille au plus 3. Ainsi, nous pouvons en temps polynomial distinguer les instances positives des instances négatives de E3SAT , c'est-à-dire le résoudre. Ce problème étant NP -complet, nous pouvons donc déduire de cette réduction que MIN COLORATION n'est pas $(4/3 - \varepsilon)$ -approximable, pour tout $\varepsilon > 0$, si $\mathbf{P} \neq \mathbf{NP}$. ■

La réduction précédente est un exemple de réduction GAP. Il s'agit de créer véritablement un saut, de séparer les instances positives des instances négatives. Plus généralement, partant d'un problème de décision D (que l'on choisira NP -complet), une réduction GAP consiste à trouver une fonction f calculable en temps polynomial transformant une instance de D en une instance d'un problème Π de maximisation de NPO de manière à ce que :

- si $I \in D^+$, alors $\text{opt}(f(I)) \geq U$;
- si $I \in D^-$, alors $\text{opt}(f(I)) \leq L$ (avec $L < U$).

On peut bien évidemment définir une réduction GAP analogue pour un problème de minimisation.

Comme précédemment, l'intérêt d'une telle réduction est qu'elle fournit un résultat négatif. En effet, si Π est approximable à rapport $\rho > L/U$, alors D est polynomial. Si D est NP -complet, nous pouvons donc conclure que le problème Π n'est pas approximable à rapport strictement supérieur à L/U si $\mathbf{P} \neq \mathbf{NP}$.

3.3.2 Généralisation : réductions préservant l'approximation

La réduction GAP consiste à réduire un problème de décision NP -complet à un problème de NPO en créant un saut. Rien ne nous empêche d'utiliser ce principe de réduction cette fois-ci d'un problème d'optimisation à un autre problème d'optimisation : sous une hypothèse d'inapproximabilité du premier problème, et si la réduction est bien construite, nous pouvons par des arguments analogues obtenir un résultat d'inapproximabilité pour le deuxième problème. Etudions l'exemple suivant (voir par exemple [56]).

Exemple 2. Considérons les problèmes MAX E3SAT et MAX STABLE . Soit une instance φ de MAX E3SAT comportant n variables et m clauses. Nous construisons le graphe suivant : pour chaque clause, nous construisons un triangle (clique de taille 3), chaque sommet du triangle correspondant à un littéral de la clause ; nous mettons une arête entre des sommets correspondant à des littéraux opposés (x_i et \bar{x}_i).

Etudions les relations entre φ et $G = f(\varphi)$. Supposons tout d'abord que nous ayons une valeur de vérité satisfaisant k clauses dans φ . Choisissons alors dans chaque

triangle correspondant à une clause satisfaite un sommet correspondant à un littéral vrai. Cet ensemble de sommets constitue un stable de G de taille k . Réciproquement, supposons que l'on ait un stable de G de taille k . Considérons comme vrais les littéraux correspondant aux sommets de ce stable (il n'y a pas de contradiction par construction), et complétons cette valeur de vérité arbitrairement. Alors cette valeur de vérité satisfait au moins k clauses.

En regroupant ces deux raisonnements, nous obtenons le résultat suivant. Si nous avons une solution S de G telle que $m(G, S) \geq \rho \text{opt}(G)$, alors nous pouvons efficacement calculer une valeur de vérité τ telle que $m(\varphi, \tau) \geq m(G, S) \geq \rho \text{opt}(G) \geq \rho \text{opt}(\varphi)$. Autrement dit, on déduit d'un algorithme ρ -approché pour MAX STABLE un algorithme ρ -approché pour MAX E3SAT . Ainsi, le résultat d'inapproximabilité de HÅSTAD s'étend à MAX STABLE : MAX STABLE n'est pas approximable à rapport $8/7 - \varepsilon$, pour tout $\varepsilon > 0$, si $\mathbf{P} \neq \mathbf{NP}$. ■

Ce type de réductions, appelées réductions préservant l'approximation, permettant d'obtenir des résultats négatifs, sera l'objet des prochaines sections.

3.3.3 Quelques mots sur PCP

La technique de la réduction GAP et celle des réductions préservant l'approximation ont été développées assez rapidement. Si elles ont permis d'obtenir quelques résultats négatifs (exemple 1), la plupart des questions concernant les problèmes les plus classiques demeuraient ouvertes. Les sauts entre résultats positifs et résultats négatifs restaient parfois immenses, pour des problèmes comme MAX STABLE par exemple. Il a fallu attendre les années 1990 pour que soient résolues bon nombre de ces questions. L'immense avancée est venue d'une nouvelle caractérisation de \mathbf{NP} par les systèmes de preuves probabilistes. Des liens entre \mathbf{NP} , systèmes de preuves probabilistes et approximation sont mis à jour par FEIGE *et al.* en 1991 dans [35]. ARORA *et al.* ([3]) ont ensuite prouvé le théorème **PCP**, proposant une nouvelle caractérisation de \mathbf{NP} . De ce théorème, on peut répondre par la négative à des questions centrales de l'approximabilité à l'époque : MAX 3SAT admet-il un schéma d'approximation?, MAX STABLE est-t-il dans \mathbf{APX} ?, etc. Ce théorème a ensuite été affiné et adapté pour obtenir des résultats d'inapproximabilité forts pour de nombreux problèmes classiques. Citons notamment les travaux de HÅSTAD (sur MAX STABLE et MAX E3SAT notamment, [40, 39]) et Feige (sur MIN SET COVER , [34]).

Nous ne détaillerons pas ici les systèmes de preuves probabilistes et le théorème **PCP**, qui dépassent le cadre de cet article. Néanmoins, nous allons en exposer très brièvement les idées principales, et invitons le lecteur à consulter l'ouvrage d'AUSIELLO *et al.* [7] (voir aussi [56]) pour une présentation et une preuve du théorème **PCP**, ainsi que l'excellent état de l'art de TREVISAN ([61], traduit en français dans [62]) ou celui d'ARORA et LUND ([2]) pour un exposé du développement des systèmes **PCP** et des résultats d'inapproximabilité auxquels ils ont conduit.

Le théorème **PCP** se base sur un rapprochement de la définition de \mathbf{NP} par l'existence de certificat et de celle d'algorithmes aléatoires. En effet, les problèmes de \mathbf{NP} sont ceux pour lesquels il existe un certificat de taille polynomiale tel que l'on peut

vérifier en temps polynomial que ce certificat constitue une preuve que l'instance est positive. Nous pouvons alors envisager de remplacer cet algorithme déterministe de vérification par un algorithme aléatoire. Un tel algorithme a en entrée l'instance du problème et le certificat, ainsi qu'un vecteur de bits aléatoires. Le fonctionnement de l'algorithme est un peu particulier : il ne va avoir accès qu'à une partie de la preuve, cette partie étant calculée à partir des bits aléatoires. Ensuite, il devra rendre son verdict en ayant lu seulement la partie de la preuve correspondante. Nous désirons bien évidemment avoir un algorithme aléatoire de vérification qui se trompe le moins possible.

Nous pouvons alors définir des classes de problèmes possédant des propriétés analogues concernant l'existence d'un tel vérificateur. La classe $\mathbf{PCP}[r(\mathbf{n}),q(\mathbf{n})]$ est constituée de l'ensemble des problèmes $D = (D^+, D^-)$ pour lesquels il existe un algorithme A aléatoire vérificateur de preuve fonctionnant en temps polynomial et tel que :

- le vecteur de bits aléatoires est de taille au plus $r(n)$;
- A a accès à au plus $q(n)$ bits du certificat ;
- si l'instance est positive, alors il existe un certificat (de taille polynomiale) pour lequel A accepte avec probabilité 1 ;
- si l'instance est négative, alors, pour tout certificat (de taille polynomiale), A accepte avec une probabilité inférieure à $1/2$.

Il s'agit alors de mettre en relation les classes \mathbf{PCP} et les classes de complexité usuelles. Le théorème fondamental consiste en la caractérisation suivante de la classe \mathbf{NP} .

Théorème 1. (*Théorème PCP*, [3]) $\mathbf{NP} = \mathbf{PCP}[\mathbf{O}(\log(\mathbf{n})), \mathbf{O}(1)]$.

Ce théorème revêt une importance majeure de part sa relation avec l'inapproximabilité des problèmes d'optimisation discrète, puisque, rappelons-le, un corollaire affirme que $\mathbf{MAX\ 3SAT}$ et $\mathbf{MAX\ STABLE}$ n'admettent pas de schéma d'approximation. Pour ce deuxième problème, la combinaison du corollaire avec un ancien résultat ([36]) affirmant que, à moins que $\mathbf{P} = \mathbf{NP}$, soit $\mathbf{MAX\ STABLE}$ admet un schéma d'approximation, soit il n'appartient pas en \mathbf{APX} , implique immédiatement sa non-approximabilité à rapport constant.

3.4 Logique, complexité optimisation et approximation

La théorie de la complexité se fonde sur la théorie des langages (cf. [49]). Une approche alternative, basée sur la définition logique de \mathbf{NP} par FAGIN ([33]) a été développée et employée de manière fertile dans la théorie de l'approximation polynomiale. Nous présentons dans ce paragraphe des résultats sur cette approche. Ils se basent sur les travaux de PAPADIPITRIOU et YANNAKAKIS ([55]), exhibant des liens forts entre logique, complexité et approximation.

Nous présentons d'abord une caractérisation en termes de logique de la classe \mathbf{NP} . FAGIN, en essayant de résoudre une question ouverte en logique³, a montré que

3. Le complément d'un spectre généralisé est-il toujours un spectre généralisé?

répondre à cette question était équivalent à déterminer si **P** est égal à **NP**. Ce faisant, il a en fait caractérisé les problèmes exprimés sous forme logique appartenant à **NP**.

Exemple 3. Etant donné un graphe G , le problème 3-COLORATION consiste à déterminer si G est 3-coloriable. Ce problème est bien évidemment dans **NP**. Un graphe G est donc la donnée d'un ensemble de sommets $V = \{v_1, \dots, v_n\}$ (que l'on appellera univers en logique) et d'un ensemble E d'arêtes, que l'on peut représenter sous la forme d'une relation binaire sur V^2 ($E(x,y)$ signifie que l'arête (x,y) existe). Une 3-coloration d'un graphe G est une partition des sommets en trois stables S_1, S_2 et S_3 . Représentant un ensemble V' de sommets comme une relation unaire sur V ($V'(v)$ signifie que $v \in V'$), alors nous pouvons caractériser les graphes 3-coloriables comme ceux pour lesquels la formule suivante est vraie :

$$\begin{aligned} \exists S_1 \exists S_2 \exists S_3 & \left(\forall x S_1(x) \vee S_2(x) \vee S_3(x) \right) \\ & \wedge \left(\forall x (\neg S_1(x) \wedge \neg S_2(x)) \vee (\neg S_1(x) \wedge \neg S_3(x)) \right. \\ & \quad \left. \vee (\neg S_2(x) \wedge \neg S_3(x)) \right) \\ & \wedge \left(\forall x \forall y \left((S_1(x) \wedge S_1(y)) \vee (S_2(x) \wedge S_2(y)) \right. \right. \\ & \quad \left. \left. \vee (S_3(x) \wedge S_3(y)) \right) \right) \\ & \Rightarrow \neg G(x,y) \end{aligned}$$

En effet, les ensembles S_1, S_2 et S_3 représentent les trois couleurs. La première ligne indique que chaque sommet doit recevoir au moins une couleur, la deuxième que chaque sommet reçoit au plus une couleur. Quant à la dernière ligne, elle traduit le fait que deux sommets adjacents ne doivent pas recevoir la même couleur. ■

Le problème de décision 3-COLORATION s'exprime donc sous la forme d'une formule logique du second ordre, les instances positives étant celles qui satisfont cette formule. Cet exemple traduit en fait une caractéristique générale à tous les problèmes de **NP**.

De manière à formaliser cette caractérisation, soient $n \in \mathbb{N}$ et $\sigma = (\sigma_1, \dots, \sigma_n) \subset \mathbb{N}^n$. Une σ -structure est un $(n+1)$ -uplet (A, P_1, \dots, P_n) où : A est un ensemble fini (non vide) appelé *univers* et P_i est un prédicat d'arité σ_i sur A . Le théorème de FAGIN affirme que pour tout problème $D = (D^+, D^-)$ de **NP**, il existe $\sigma \in \mathbb{N}^{n_1}$, $\mu \in \mathbb{N}^{n_2}$ et une formule du premier ordre Φ tels que l'on peut représenter le problème D de **NP** de la manière suivante :

- les instances I de Π sont représentées par des σ -structures I_σ ;
- l'instance I est positive si et seulement s'il existe une μ -structure \mathcal{S} telle que $(I_\sigma, \mathcal{S}) \models \Phi(I_\sigma, \mathcal{S})$.

Nous renvoyons le lecteur également au livre de PAPADIMITRIOU ([54]) pour une introduction et une démonstration détaillées de ce résultat.

Partant de cette caractérisation donnée par FAGIN et de l'importance majeure des problèmes logiques (de satisfiabilité) en optimisation et en particulier en approximation, il paraît naturel de s'interroger sur d'éventuels liens entre problèmes exprimés/exprimables sous forme logique, optimisation et approximation. Le premier travail dans ce sens a été proposé par PAPADIMITRIOU et YANNAKAKIS ([55]). Un des buts de cet article a été de définir des classes syntaxiques naturelles définissant des problèmes de **NPO** ayant des propriétés d'approximation.

Pour tout $n \in \mathbb{N}$, nous appellerons Σ_n (resp., Π_n) l'ensemble des formules logiques du premier ordre comportant, en notation préfixe, une succession de n blocs de quantificateurs, alternant des blocs de \forall et de \exists commençant par \exists (resp., par \forall). FAGIN exprime les problèmes de **NP** par une formule du second ordre. Considérons alors un problème D de **NP** exprimé par la formule $\exists S(I, S) \models \Phi(I, S)$. La formule Φ est une formule du premier ordre. Si elle commence par un quantificateur \forall , nous pouvons alors définir un problème d'optimisation Π_D « correspondant » à D :

$$\max_S |\{x : (I, S) \models \Phi'(x, I, S)\}|$$

où Φ' est la formule obtenue à partir de Φ en retirant le premier quantificateur (x étant donc un k -uplet d'éléments de l'univers de I).

Définition 2. ([55]) **Max-SNP** est l'ensemble des problèmes de maximisation définis par $\max_S |\{x : (I, S) \models \varphi(x, I, S)\}|$, où φ est une formule du premier ordre *sans quantificateur*. Il s'agit plus précisément de trouver une structure S^* maximisant l'expression ci-dessus. **Max-NP** est l'ensemble des problèmes de maximisation définis par $\max_S |\{x : (I, S) \models \exists y \varphi(x, y, I, S)\}|$, où φ est une formule du premier ordre *sans quantificateur*. Il s'agit ici aussi de trouver une structure S^* maximisant l'expression ci-dessus. ■

La définition 2 ne rentre pas vraiment dans le cadre formel des problèmes de **NPO**. Il semble cependant que par cette définition les auteurs de [55] ont voulu signifier que les structures S représentaient d'une certaine manière les solutions réalisables de l'instance du problème. Nous retrouvons alors le cadre habituel en considérant la fonction objectif $m(I, S)$: $m(I, S) = |\{x : (I, S) \models \exists y \varphi(x, y, I, S)\}|$.

Notons que beaucoup de problèmes classiques appartiennent à ces deux classes **Max-SNP** et **Max-NP**. Par exemple, **MAX 3SAT**, **MAX STABLE- B** (restriction de **MAX STABLE** aux graphes de degré borné par B) et **MAX COUPE**⁴ sont dans **Max-SNP**, et **MAX SAT** est dans **Max-NP**. Le premier lien entre les classes syntaxiques **Max-SNP** et **Max-NP**, et l'approximation correspond au théorème suivant.

Théorème 2. ([55]) *Tout problème de **Max-SNP** et de **Max-NP** est approximable à rapport constant. Autrement dit, **Max-SNP** et **Max-NP** sont incluses dans **APX**.*

4. Etant donné un graphe, on cherche à déterminer un sous-ensemble de sommets V' maximisant le nombre d'arêtes ayant une extrémité et une seule dans V' .

4 Réductions et complétude

Nous abordons maintenant un des sujets qui se situent au cœur de l'intersection complexité – approximation, la structuration des classes d'approximation par l'introduction de notions de complétude pour les classes d'approximabilité présentées dans le paragraphe 3.2.

Revenons un instant sur l'exemple 2 de la section 3.3.2. Nous avons présenté une réduction de MAX E3SAT à MAX STABLE de manière à ce que l'on puisse transférer un résultat d'approximabilité de MAX STABLE à MAX E3SAT . Cette réduction est une illustration du concept de réductions préservant l'approximation. Ces réductions (il en existe plusieurs types) traduisent l'idée selon laquelle un problème est plus difficile qu'un autre à résoudre, non plus en termes de polynomialité de la résolution optimale (telle était le cas des réductions de KARP et de TURING vues dans la section 2), mais en termes de résolution approchée. De cette notion de réduction, parallèlement à celle déjà largement évoquée de NP -complétude, nous pouvons alors définir des notions de complétude qui structurent, à la manière de la réduction de Karp pour NP , les classes de problèmes de NPO .

4.1 Réductions préservant l'approximation

Revenons à nouveau sur l'exemple 2 de la réduction de MAX E3SAT à MAX STABLE proposé à la section 3.3.2. Le principe de cette réduction est de transformer :

- une instance φ de MAX E3SAT en une instance $G = f(\varphi)$ de MAX STABLE ;
- une solution réalisable S de G en une solution réalisable $g(\varphi, S) = \sigma$ de φ , de manière à ce que les rapports d'approximation des solutions S et σ soient reliés de la manière suivante : $\rho(G, S) \leq \rho(\varphi, \sigma)$.

Cette réduction permet, nous l'avons vu, d'affirmer que si le problème MAX STABLE est ρ -approximable, pour une certaine constante ρ , alors le problème de départ MAX E3SAT est ρ -approximable.

De manière équivalente, nous avons formulé la contraposée : si MAX E3SAT n'est pas approximable à rapport ρ (sous une hypothèse de complexité du type $\mathbf{P} \neq \mathbf{NP}$), alors MAX STABLE n'est pas approximable à rapport ρ (sous la même hypothèse de complexité).

Cet exemple résume le principe et l'intérêt des réductions préservant l'approximation. Il s'agit donc d'exprimer que tel problème est plus « facile » (dans un sens dépendant de la réduction) à approcher que tel autre. Pour parvenir à cette conclusion, on réutilise l'idée des réductions entre problèmes de décision en déterminant en temps polynomial une solution approchée du problème initial en supposant que l'on sache résoudre convenablement de manière approchée le problème final.

Les réductions usuelles comportent donc les trois composantes suivantes :

- une manière de transformer les instances du problème initial Π en instances du problème final Π' , c'est-à-dire, une fonction $f : \mathcal{I}_\Pi \rightarrow \mathcal{I}_{\Pi'}$ calculable en temps polynomial ;

- une manière de transformer les solutions réalisables pour les instances de Π' en solutions réalisables pour les instances de Π , c'est-à-dire, une fonction g calculable en temps polynomial associant à toute instance I de Π et à toute solution réalisable x' de $f(I)$ une solution $x = g(I, x')$ de I ;
- une propriété concernant le transfert des rapports d'approximation, traduisant l'idée que si x' atteint un certain niveau d'approximation, alors $x = g(I, x')$ doit atteindre un certain niveau (éventuellement différent) d'approximation.

Si l'on dispose d'une réduction de ce type, on peut alors utiliser un algorithme approché A' pour Π' pour résoudre de manière approchée le problème Π : il suffit de transformer une instance I de Π par f , d'utiliser A' pour obtenir une solution réalisable de $f(I)$, puis g pour obtenir une solution réalisable de I . Autrement dit, nous avons défini l'algorithme A renvoyant $A(I) = g(I, A'(f(I)))$. Cet algorithme est évidemment polynomial si A' l'est. Les résultats d'approximabilité obtenus dépendent du troisième item de la réduction.

Afin d'éclaircir cette introduction aux réductions préservant l'approximation, donnons dès à présent deux exemples de réductions fréquemment utilisées : la réduction L , pour le rapport standard, et la réduction AF pour le rapport différentiel.

Définition 3. ([55]) Un problème $\Pi \in \mathbf{NPO}$ se réduit à un problème $\Pi' \in \mathbf{NPO}$ sous la réduction L s'il existe deux fonctions f et g calculables en temps polynomial et deux constantes α et β telles que :

1. pour tout $I \in \mathcal{I}_\Pi$, $f(I) \in \mathcal{I}_{\Pi'}$; de plus, $\text{opt}_{\Pi'}(f(I)) \leq \alpha \text{opt}_\Pi(I)$;
2. pour tout $I \in \mathcal{I}_\Pi$ et tout $x \in \text{sol}(f(I))$, $g(I, x) \in \text{sol}(\mathcal{I}_\Pi)$; de plus :

$$\left| m_\Pi(I, g(I, x)) - \text{opt}_\Pi(I) \right| \leq \beta \left| m_{\Pi'}(f(I), x) - \text{opt}_{\Pi'}(f(I)) \right| \blacksquare$$

Remarquons que les deux conditions relatives à α et β impliquent : $|\rho_\Pi(I, g(I, x)) - 1| \leq \alpha\beta|\rho_{\Pi'}(f(I), x) - 1|$. Ainsi, si par exemple Π et Π' sont deux problèmes de maximisation, si x est une solution ρ' approchée pour $f(I)$, alors $\rho_\Pi(I, g(I, x)) \geq 1 - \beta\alpha(1 - \rho')$. Ceci permet d'obtenir la propriété fondamentale suivante (valable même si les problèmes considérés ne sont pas des problèmes de maximisation) : *si $\Pi' \in \mathbf{PTAS}$ et si $\Pi \leq_L \Pi'$, alors $\Pi \in \mathbf{PTAS}$.*

Exemple 4. $\text{MAX STABLE-}B$ se L -réduit à MAX 2SAT . Le problème $\text{MAX STABLE-}B$ est la restriction du problème MAX STABLE aux graphes de degré au plus B , le problème MAX 2SAT étant la restriction du problème MAX SAT aux instances où les clauses sont de taille au plus 2. Considérons une instance de $\text{MAX STABLE-}B$, c'est-à-dire, un graphe $G(V, E)$ de degré au plus B . Nous allons considérer l'instance φ de MAX 2SAT suivante : à chaque sommet v_i nous associons une variable x_i ; nous considérons la clause $\bar{x}_i \vee \bar{x}_j$ pour chaque arête (v_i, v_j) , et la clause x_i pour chaque sommet v_i . Nous venons ainsi de définir la fonction f de notre réduction L .

Pour définir la fonction g , considérons une valeur de vérité τ . Appliquons alors la règle suivante à τ : dès qu'une clause $\bar{x}_i \vee \bar{x}_j$ n'est pas vérifiée, changeons la valeur de x_i (fixons-la à *faux*). On obtient au final une valeur de vérité τ' dont on voit

aisément qu'elle satisfait au moins autant de clauses que τ (car c'est vrai à chacune des étapes). Considérons alors l'ensemble de sommets V' constitué des variables valant *vrai* dans τ' . L'ensemble V' est un stable, car toutes les clauses provenant des arêtes sont vérifiées, et voilà la fonction g définie.

Il reste à montrer que la réduction proposée est une réduction L. Pour cela, remarquons que $m(\varphi, \tau) \leq m(\varphi, \tau') = |E| + |V'|$. Sachant que le degré du graphe est borné par B , nous savons d'une part que $|E| \leq nB/2$, d'autre part qu'il existe un stable de taille au moins $n/(B+1)$. A l'aide de ces deux observations, nous déduisons que $|E| \leq B(B+1) \text{opt}(G)/2$. Ainsi: $\text{opt}(\varphi) = |E| + \text{opt}(G) \leq ((B(B+1)/2) + 1) \text{opt}(G)$. La condition 1 de la définition 3 est donc vérifiée, en prenant $\alpha = B(B+1)/2 + 1$. D'autre part: $\text{opt}(\varphi) - m(\varphi, \tau) \geq \text{opt}(\varphi) - m(\varphi, \tau') = \text{opt}(G) - m(G, V')$. Ceci n'est autre que la condition 2 requise par la réduction L, avec $\beta = 1$. ■

La réduction AF se situe quant à elle dans le cadre de l'approximation différentielle. Elle relate l'existence d'une relation affine entre deux problèmes, ce qui permet d'exploiter la stabilité du rapport différentiel sous ce type de transformations.

Définition 4. Un problème $\Pi \in \mathbf{NPO}$ se réduit à un problème $\Pi' \in \mathbf{NPO}$ sous la réduction AF s'il existe deux fonctions f et g calculables en temps polynomial et deux constantes $\alpha \neq 0$ et β telles que :

- pour tout $I \in \mathcal{I}_\Pi$, $f(I) \in \mathcal{I}_{\Pi'}$;
- pour tout $I \in \mathcal{I}_\Pi$ et tout $x \in \text{sol}(f(I))$, $g(I, x) \in \text{sol}(\mathcal{I}_{\Pi'})$; de plus : $m_\Pi(I, g(I, x)) = \alpha m_{\Pi'}(f(I), x) + \beta$;
- pour tout I , $g(I, \cdot)$ est surjective ;
- les problèmes ont le même but (maximisation ou minimisation) si $\alpha > 0$, de buts différents si $\alpha < 0$. ■

Grâce à la surjectivité de la fonction $g(I, \cdot)$, les valeurs opt et ω vérifient $\text{opt}_\Pi(I) = \alpha \text{opt}_{\Pi'}(f(I)) + \beta$ et $\omega_\Pi(I) = \alpha \omega_{\Pi'}(f(I)) + \beta$. Ceci permet d'affirmer que : *si $\Pi' \in \mathbf{DPTAS}$ (resp., $\Pi' \in \mathbf{DAPX}$) et si $\Pi \leq_{\text{AF}} \Pi'$, alors $\Pi \in \mathbf{DPTAS}$ (resp., $\Pi \in \mathbf{DAPX}$).*

4.2 Complétude

Les réductions (dont celles préservant l'approximation) reflètent l'idée qu'un problème est plus facile à résoudre qu'un autre problème. Elles peuvent ainsi se voir comme une relation (binaire) de difficulté entre les problèmes. Rappelons qu'une relation binaire R sur un ensemble E est dite : *réflexive*, si pour tout $x \in E$, xRx , *transitive*, si pour tout $(x, y, z) \in E^3$, $xRy \wedge yRz \Rightarrow xRz$, *symétrique*, si pour tout $(x, y) \in E^2$, $xRy \Rightarrow yRx$ et *antisymétrique*, si pour tout $(x, y) \in E^2$, $xRy \wedge yRx \Rightarrow x = y$. Une relation est appelée *ordre* si elle est réflexive, transitive et antisymétrique, *préordre* si elle est réflexive et transitive, *relation d'équivalence* si elle est réflexive, transitive et symétrique.

Les réductions sont généralement (c'est le cas pour toutes les réductions connues) réflexives et transitives. Elles induisent autrement dit un *préordre partiel* sur l'ensemble

des problèmes dont il est question. Préordre et non pas ordre car elles ne sont pas généralement antisymétriques, partiel car il est parfaitement possible que deux problèmes soient incomparables au sens d'une réduction, c'est-à-dire, qu'il peut exister deux problèmes Π et Π' tels que, pour une réduction R , nous n'ayons ni $\Pi \leq_R \Pi'$ ni $\Pi' \leq_R \Pi$. Ces deux derniers points sont là encore vérifiés pour toutes les réductions que nous aborderons.

Partant d'une relation réflexive et transitive, nous pouvons, de manière très classique, définir la relation d'équivalence associée. Soit R une réduction (réflexive et transitive). On définit la relation d'équivalence \equiv_R de la manière suivante : $\Pi \equiv_R \Pi'$ si et seulement si $\Pi \leq_R \Pi'$ et $\Pi' \leq_R \Pi$. On dit alors que Π et Π' sont équivalents sous la réduction R .

Etant donné un ensemble de problèmes C et une réduction R , il est alors naturel de rechercher s'il existe des éléments maximaux dans le préordre induit par R , autrement dit s'il existe des problèmes Π de C tels que tout problème Π' de C se réduit à Π . De tels éléments maximaux sont appelés, dans le langage de la complexité, problèmes complets.

Soient une classe de problèmes C et une réduction R . Un problème $\Pi \in C$ est dit C -complet (sous la réduction R) si : pour tout $\Pi' \in C$, $\Pi' \leq_R \Pi$. Un problème C -complet (sous une réduction R) est alors, au sens de la réduction R , le problème le plus dur à résoudre de la classe C . Dans le cas de la \mathbf{NP} -complétude par exemple, les problèmes \mathbf{NP} -complets (sous la réduction de \mathbf{KARP} par exemple) sont les plus durs à résoudre puisque si l'on en résolvait un en temps polynomial, tous les problèmes de \mathbf{NP} seraient polynomiaux.

Concernant les relations de difficultés entre problèmes, il peut être également intéressant d'exhiber des problèmes auxquels tous ceux d'une classe C se réduisent, même si ces problèmes ne sont pas dans la classe de départ. De tels problèmes sont alors dits (par analogie là encore avec les problèmes de \mathbf{NP}) C -difficiles sous la réduction R .

Soient une classe de problèmes C et une réduction R . Un problème Π est dit C -difficile (sous la réduction R) si : pour tout $\Pi' \in C$, $\Pi' \leq_R \Pi$. Autrement dit, un problème Π est C -complet si et seulement s'il appartient à C et est C -difficile.

Enfin, d'un point de vue purement structurel, il peut être intéressant de déterminer la clôture d'une classe sous une réduction particulière, la clôture d'une classe étant définie comme l'ensemble des problèmes pouvant se réduire à un problème de la classe en question. Il s'agit donc de l'ensemble des problèmes plus « faciles » à résoudre qu'au moins un des problèmes de la classe. Soient C une classe de problèmes et R une réduction. La clôture de C sous R est : $\bar{C}^R = \{\Pi : \exists \Pi' \in C, \Pi \leq_R \Pi'\}$.

4.3 Liens avec l'approximation

Nous avons déjà vu, dans l'exemple 2 notamment, que la notion de réduction préservant l'approximation est fortement liée à celle d'approximation. Cette relation se révèle d'ailleurs souvent plus utile pour montrer des résultats d'inapproximabilité. Généralisant en un certain sens la réduction \mathbf{GAP} , elle permet de transférer des résultats d'inapproximabilité d'un problème Π à un problème Π' .

Il s'agit là de résultats issus d'une relation exhibée entre deux problèmes particuliers. Cependant, nous pouvons également nous positionner à un niveau plus global (au niveau de classes de problèmes), et voir comment cette notion de réduction peut être liée à l'approximation. Pour cela, nous allons mettre en relation les notions de complétudes induites par des réductions possédant certaines propriétés, avec celle de hiérarchie de classes d'approximation.

Reprenons un instant l'exemple de la réduction L , introduite à la section 4.1. Comme énoncé après sa définition, si un problème Π' admet un schéma d'approximation et si un problème Π se L -réduit à Π' , nous déduisons l'existence d'un schéma pour Π . Nous dirons dorénavant que la réduction L *préserve l'appartenance à \mathbf{PTAS}* . Imaginons alors une classe qui contient \mathbf{PTAS} , disons \mathbf{APX} par exemple. Supposons que l'on ait trouvé un problème Π \mathbf{APX} -complet sous la réduction L . Si Π admet un schéma, alors puisque la réduction L préserve l'appartenance à \mathbf{PTAS} , nous déduisons l'existence d'un schéma pour tous les problèmes se L -réduisant à Π , donc en particulier pour tous les problèmes de \mathbf{APX} . Autrement dit, $\Pi \in \mathbf{PTAS} \Rightarrow \mathbf{APX} = \mathbf{PTAS}$. Puisque nous savons que, sous l'hypothèse $\mathbf{P} \neq \mathbf{NP}$, $\mathbf{PTAS} \subsetneq \mathbf{APX}$, nous aboutissons à la conclusion que, sous cette même hypothèse $\mathbf{P} \neq \mathbf{NP}$, $\Pi \notin \mathbf{PTAS}$.

C'est cette relation entre classes de problèmes et (in)approximabilité qui motive principalement l'introduction de la notion de complétude précédemment définie. Ce schéma de raisonnement se généralise. Soit \mathbf{C} une classe de problèmes. On dit qu'une réduction R préserve l'appartenance à \mathbf{C} si pour tous problèmes Π et Π' : $\Pi \in \mathbf{C}$ et $\Pi' \leq_R \Pi$, alors $\Pi' \in \mathbf{C}$.

Proposition 2. *Soient de classes de problèmes \mathbf{C} et \mathbf{C}' avec $\mathbf{C}' \subsetneq \mathbf{C}$. Si un problème Π est \mathbf{C} -complet sous une réduction préservant l'appartenance à \mathbf{C}' , alors $\Pi \notin \mathbf{C}'$.*

Remarquons que, bien entendu, si l'inclusion stricte des classes est soumise à une hypothèse (de complexité notamment), la conclusion $\Pi \notin \mathbf{C}'$ est valable sous la même hypothèse de complexité. Le parallèle avec la complexité des problèmes de décision est évidemment immédiat. La propriété fondamentale de la réduction de KARP (ou de celle de TURING) est qu'elle préserve l'appartenance à \mathbf{P} . Le deuxième des énoncés précédents se traduit alors par l'énoncé classique établissant que les problèmes \mathbf{NP} -complets ne sont pas dans \mathbf{P} sauf si \mathbf{P} et \mathbf{NP} coïncident. Poursuivons un instant ce parallèle. Comme exposé dans la section 2, un résultat de LADNER ([47]) établit l'existence de problèmes dits intermédiaires (si $\mathbf{P} \neq \mathbf{NP}$) sous la réduction de KARP. Un problème de \mathbf{NP} est intermédiaire s'il n'est ni dans \mathbf{P} (si $\mathbf{P} \neq \mathbf{NP}$), ni \mathbf{NP} -complet. Cela revient à dire qu'il n'y a pas que deux niveaux dans la hiérarchie de difficulté induite par la réduction de KARP (problèmes polynomiaux et problèmes \mathbf{NP} -complets), ou encore qu'il n'est pas équivalent de dire qu'un problème est \mathbf{NP} -complet ou qu'il n'est pas polynomial si $\mathbf{P} \neq \mathbf{NP}$. Il apparaît alors naturel de se poser la même question pour la complétude dans les classes d'approximation. Ainsi, peut-on dire qu'un problème dans \mathbf{APX} qui n'admet pas de schéma est \mathbf{APX} -complet sous la réduction L ? C'est la problématique de la recherche de problèmes intermédiaires.

Soient deux classes de problèmes \mathbf{C} et \mathbf{C}' avec $\mathbf{C}' \subseteq \mathbf{C}$, et une réduction R préservant l'appartenance à \mathbf{C}' . On dit qu'un problème est *\mathbf{C} -intermédiaire (sous la réduction*

tion R , relativement à \mathcal{C}') s'il n'est ni \mathbf{C} -complet sous R , ni dans \mathcal{C}' sous l'hypothèse $\mathcal{C}' \not\subseteq \mathbf{C}$.

Les sections 5, 6 et 7 qui suivent décrivent les trois techniques utilisées pour obtenir des résultats de complétude pour les classes d'approximabilité.

5 Complétude sans utiliser PCP

Nous exposons dans cette section des résultats de complétude pour les classes **NPO** (section 5.1), **APX** (section 5.2) et **PTAS** (section 5.3). Ces résultats de complétude s'obtiennent en se basant sur une même technique consistant à reprendre et à modifier la preuve du théorème de COOK en l'adaptant dans le contexte « optimisation » et sans utiliser le théorème **PCP** (théorème 1, section 3.3.3).

5.1 NPO-complétude

Comme nous l'avons déjà expliqué, pour définir une notion de complétude, nous devons tout d'abord définir une réduction préservant l'approximation. L'idée qui semble la plus naturelle (et elle apparaît effectivement dès l'article de ORPONEN et MANNILA [53]) consiste à dire que nous voulons récupérer, grâce à la fonction g de notre réduction, une solution $g(I,x)$ au moins aussi bonne que la solution x , au sens des rapports d'approximation bien évidemment.

Définition 5. ([53]) Soient Π et Π' deux problèmes de maximisation de **NPO**. On dit que Π se réduit à Π' sous la *réduction stricte* s'il existe deux fonctions f et g calculables en temps polynomial telles que :

- pour tout $I \in \mathcal{I}_\Pi$, $f(I) \in \mathcal{I}_{\Pi'}$;
- pour tout $I \in \mathcal{I}_\Pi$ et tout $x \in \text{sol}(f(I))$, $g(I,x) \in \text{sol}(\mathcal{I}_\Pi)$;
- pour tout $I \in \mathcal{I}_\Pi$ et tout $x \in \text{sol}(f(I))$, $\rho_\Pi(I,g(I,x)) \geq \rho_{\Pi'}(f(I),x)$. ■

On peut définir de manière analogue la réduction stricte lorsqu'il s'agit de problèmes de minimisation.

Cette réduction combine l'avantage de préserver l'appartenance à des classes d'approximation classiques (**APX** et **PTAS** par exemple) et l'inconvénient d'être très restrictive, puisqu'elle n'autorise aucune dégradation de la qualité de la solution. Néanmoins, des résultats (partiels) de complétude ont été obtenus dans la classe **NPO** sous la réduction stricte. Plus précisément, définissons les problèmes **MAX SAT PONDÉRÉ** et **MIN SAT PONDÉRÉ**. Dans ces deux problèmes, on nous fournit une instance de **SAT** sur m clauses et n variables, ainsi qu'un poids $w(x_i)$ sur chaque variable x_i . Les solutions réalisables sont les valeurs de vérité vérifiant φ . La valeur d'une telle valeur de vérité τ est la somme des poids des variables mises à *vrai*, c'est-à-dire $\sum_{i=1}^n w(x_i)\tau(x_i)$. Il s'agit alors bien entendu de maximiser (dans **MAX SAT PONDÉRÉ**) ou de minimiser (dans **MIN SAT PONDÉRÉ**) cette valeur pour toutes les solutions réalisables. Néanmoins, ainsi définis, ces deux problèmes n'entrent pas dans le cadre de la définition de **NPO** donnée à la section 2. En effet, trouver une solution réalisable (quand il en

existe) revient à résoudre un problème SAT, ce qui n'est donc pas polynomialement faisable si $\mathbf{P} \neq \mathbf{NP}$. Pour remédier à cela, nous rajoutons simplement une solution réalisable triviale. Dans MAX SAT PONDÉRÉ, nous considérons que la solution mettant toutes les variables à *faux* est réalisable (et de valeur 0) ; dans MIN SAT PONDÉRÉ, nous considérons que la solution mettant toutes les variables à *vrai* est réalisable (et de valeur $\sum_{i=1}^n w(x_i)$).

Théorème 3. ([53]) MIN SAT PONDÉRÉ est **Min-NPO-complet** sous la réduction stricte.

Preuve. ([53]) Nous allons donner la preuve proposée par ORPONEN et MANNILA. Celle-ci se base sur la preuve du théorème de COOK, c'est-à-dire, la construction d'une formule sous forme conjonctive normale « simulant » le fonctionnement d'une machine de TURING non déterministe. Nous présentons cette démonstration car l'idée sur laquelle elle se base se retrouve dans la plupart des preuves de complétude de cette section.

Considérons donc un problème Π de minimisation de **NPO**. Soit p un polynôme bornant la taille des solutions réalisables, c'est-à-dire un polynôme tel que pour toute instance I de Π et pour toute solution réalisable x de I , $|x| \leq p(|I|)$. Considérons alors la machine non déterministe suivante :

Machine M
sur l'entrée I :
si I n'est pas une instance de Π , alors rejeter ;
générer une solution x de taille au plus $p(I)$;
si x n'est pas une solution réalisable, alors rejeter ;
écrire x ;
écrire $m(I, x)$;
accepter.

Partant de cette machine de TURING et d'une instance I , nous pouvons grâce au théorème de COOK construire une formule φ_I instance de SAT telle que les valeurs de vérité satisfaisant φ_I (s'il y en a) « correspondent » aux calculs acceptant de la machine de TURING. Considérons alors les cases c_k, \dots, c_0 de la machine où est écrite $m_{\Pi}(I, x)$ (i.e. $m_{\Pi}(I, x) = \sum_{i:c_i=1} 2^i$). Ces cases correspondent dans la preuve du théorème de Cook à des variables x_k, \dots, x_0 dans φ_I . Nous pouvons maintenant définir notre réduction : $f(I) = (\varphi_I, W)$ où $w(x_i) = 2^i$, $i = 0, \dots, k$, toutes les autres variables ayant un poids nul ; $g(I, \tau)$ vaut la solution écrite par la machine de Turing (codée dans τ) si τ satisfait φ_I , une solution quelconque sinon.

Ainsi, si τ satisfait φ alors la valeur de τ et celle de $g(I, \tau)$ sont les mêmes ; si τ ne satisfait pas φ , τ est de valeur $2^{k+1} - 1$, donc de valeur supérieure ou égale à $g(I, \tau)$. Les optima coïncidant, nous obtenons donc une réduction stricte de tout problème de minimisation de **NPO** à MIN SAT PONDÉRÉ. ■

Un résultat analogue a été montré pour les problèmes de maximisation ([10]) : MAX SAT PONDÉRÉ est **Max-NPO-complet** sous la réduction stricte.

Il est à noter qu'ORPONEN et MANNILA prouvent la **Min-NPO-complétude** de plusieurs autres problèmes : MIN 3SAT PONDÉRÉ, restriction de MIN SAT PONDÉRÉ

à une formule comportant des clauses de taille au plus 3, le voyageur de commerce MIN TSP et la programmation linéaire en variables binaires MIN PLO-1 . Indiquons en outre que l'on trouve également la démonstration de ces résultats de complétude (ou du moins l'idée principale de ces preuves), exprimées dans des cadres quelque peu différents, dans [46] et [57].

Vient alors naturellement, à la suite de ces résultats, la question de déterminer si l'un des problèmes **Min-NPO**- ou **Max-NPO**-complets est complet pour la classe **NPO** entière. Malheureusement, aucun résultat de **NPO**-complétude n'a été obtenu sous la réduction stricte. En réalité, cela semble provenir d'une dissymétrie entre les problèmes de maximisation et de minimisation de **NPO**. Plus précisément, KOLAITIS et THAKUR montrent dans [45] que réécrire tout problème de maximisation en un problème de minimisation sur les mêmes instances de manière à ce que les optima coïncident (ou vice versa) est équivalent à montrer que $\mathbf{NP} = \mathbf{co-NP}$ ⁵. Il semble donc qu'en particulier, pour prouver un résultat de complétude pour **NPO**, il soit préférable d'utiliser une réduction qui ne préserve pas l'optimalité des solutions (c'est-à-dire, que partant d'une solution optimale on ne récupère pas, par la fonction g , une solution optimale).

Partant de ce constat, CRESCENZI *et al.* proposent dans [18] une nouvelle réduction, la réduction AP.

Définition 6. ([18]) Soient Π et Π' deux problèmes de **NPO**. On dit que Π se réduit à Π' sous la réduction AP s'il existe deux fonctions f et g calculables en temps polynomial et une constante strictement positive α telles que :

- pour tout $I \in \mathcal{I}_\Pi$ et pour tout $r > 1$, $f(I, r) \in \mathcal{I}_{\Pi'}$;
- pour tout $I \in \mathcal{I}_\Pi$, tout $r > 1$ et tout $x \in \text{sol}(f(I, r))$, $g(I, r, x) \in \text{sol}(\mathcal{I}_\Pi)$;
- pour tout $I \in \mathcal{I}_\Pi$, tout $r > 1$ et tout $x \in \text{sol}(f(I, r))$: $R_{\Pi'}(f(I, r), x) \leq r$ implique $R_\Pi(I, g(I, r, x)) \leq 1 + \alpha(r - 1)$, où R vaut le rapport d'approximation ρ pour un problème de minimisation et $1/\rho$ pour un problème de maximisation. ■

Notons que cette réduction préserve l'appartenance aux classes **APX** et **PTAS**. En revanche, nous voyons qu'elle est beaucoup moins contraignante que la réduction stricte. En particulier, faire dépendre les fonctions f et g de r permet de ne pas transformer des solutions optimales en solutions optimales, et ainsi de surpasser la difficulté précédemment évoquée. Ainsi, grâce à cette réduction, CRESCENZI *et al.* prouvent la complétude de plusieurs problèmes pour **NPO**. Plus précisément, ils montrent d'abord que MIN SAT PONDÉRÉ et MAX SAT PONDÉRÉ sont mutuellement réductibles sous la réduction AP.

Théorème 4. ([18]) MIN SAT PONDÉRÉ et MAX SAT PONDÉRÉ sont équivalents sous la réduction AP.

5. Les problèmes de décision de **co-NP** sont les problèmes « complémentaires » de ceux de **NP** : les instances positives d'un problème de **co-NP** sont les instances négatives d'un problème de **NP** (et réciproquement) ; $\mathbf{P} \subseteq \mathbf{co-NP}$; par ailleurs, la question $\mathbf{NP} = \mathbf{co-NP}$?, même réputée moins « cruciale » que $\mathbf{P} = \mathbf{NP}$?, est une autre fameuse question ouverte de la théorie de la complexité, dont il est communément considéré que la réponse est négative.

Corollaire 1. ([18]) *Tout problème **Max-NPO-complet** ou **Min-NPO-complet** sous la réduction AP est **NPO-complet** sous cette même réduction.*

En effet, soit un problème Π qui est, par exemple, **Max-NPO-complet** sous la réduction AP, et soit Π' un problème de minimisation de **NPO**. Nous avons $\Pi' \leq_{AP} \text{MIN SAT PONDÉRÉ} \leq_{AP} \text{MAX SAT PONDÉRÉ} \leq_{AP} \Pi$. Le résultat du corollaire 1 se déduit alors de la transitivité de la réduction AP.

Corollaire 2. ([18]) *Les problèmes d'optimisation **MAX SAT PONDÉRÉ**, **MIN SAT PONDÉRÉ**, **MIN 3SAT PONDÉRÉ**, **MIN TSP** et **MIN PL0-1** sont **NPO-complets** sous la réduction AP.*

Pour cela, il suffit de remarquer que la réduction stricte est un cas particulier de la réduction AP (en prenant $\alpha = 1$ et f et g qui ne dépendent pas de r), et d'utiliser le corollaire 1.

Le corollaire 2 soulève néanmoins une interrogation. Nous voyons en effet que les cinq problèmes dont il est question sont des problèmes où apparaissent des poids (sur les variables pour les trois premiers, sur les arêtes pour le quatrième, les coefficients pour le dernier), ces poids pouvant être très grands (exponentiels) en la taille de l'instance. Nous pouvons d'ailleurs remarquer que dans la preuve du théorème 3, les poids 2^i que nous mettons sur les variables sont exponentiels. On peut alors légitimement se demander si cet état de fait est fortuit ou pas. CRESCENZI *et al.* ont répondu par la négative en montrant dans [20] le théorème 5 suivant.

Théorème 5. ([20]) *Un problème de **NPO** polynomialement borné ne peut être **NPO-complet** (sous la réduction AP) sauf si la hiérarchie polynomiale est finie⁶.*

Notons néanmoins que dans [20] CRESCENZI *et al.* prouvent ce résultat sous la réduction PTAS. Cette réduction (que nous définissons dans la section 5.2) étant plus générale que la réduction AP, le résultat reste donc valable sous la réduction AP.

Ce dernier résultat traduit donc une difficulté d'établir une réduction générique de toute la classe **NPO** à un problème polynomialement borné. Qu'en est-il alors lorsque nous nous restreignons, pour les problèmes de départ, à des problèmes polynomialement bornés? Pouvons-nous obtenir des problèmes complets pour la classe **NPO-PB** des problèmes de **NPO** polynomialement bornés? Le travail de CRESCENZI *et al.* dans [18] répond à cette interrogation.

Considérons les problèmes **MIN PL0-1-PB** et **MAX PL0-1-PB**, restrictions des problèmes **MIN PL0-1** et **MAX PL0-1** aux cas où les coefficients de la fonction objectif sont binaires. Ces problèmes sont évidemment polynomialement bornés. **MIN PL0-1-PB** et **MAX PL0-1-PB** sont **NPO-PB-complets** sous la réduction AP ([18]).

Notons que, comme dans le cas de la complétude dans **NPO**, CRESCENZI *et al.* obtiennent ce résultat en montrant l'équivalence de ces deux problèmes et se basant sur des résultats de complétude partielle antérieurs réalisés par BERMAN et SCHNITGER ([16]) et KANN ([42]).

6. Traduction de l'anglais *polynomial hierarchy collapses* ([54]).

5.2 APX-complétude

Après avoir abordé la complétude pour la classe des problèmes de **NPO**, nous allons présenter les principaux résultats concernant la classe d'approximation la plus étudiée, la classe **APX**. Cette présentation est articulée en trois parties : des résultats de complétude proprement dits (partie principale), d'autres concernant l'existence de problèmes intermédiaires, quelques mots enfin sur les limites de cette approche vis-à-vis du théorème **PCP**.

5.2.1 Réductions et complétude

Les premiers résultats concernant la complétude dans **APX** ont été proposés par CRESCENZI et PANCONESI dans [19]. En accord avec l'intérêt de la notion de complétude précédemment évoquée, les auteurs définissent tout d'abord une réduction préservant l'appartenance à **PTAS**, la réduction **P**.

Définition 7. ([19]) Soient Π et Π' deux problèmes de maximisation de **NPO**. On dit que Π se réduit à Π' sous la réduction **P** s'il existe deux fonctions f et g calculables en temps polynomial et une fonction c telles que :

- pour tout $I \in \mathcal{I}_\Pi$, $f(I) \in \mathcal{I}_{\Pi'}$;
- pour tout $I \in \mathcal{I}_\Pi$ et tout $x \in \text{sol}(f(I))$, $g(I,x) \in \text{sol}(\mathcal{I}_\Pi)$;
- $c :]0,1[\rightarrow]0,1[$;
- pour tout $I \in \mathcal{I}_\Pi$, tout $x \in \text{sol}(f(I))$ et tout $\varepsilon \in]0,1[$: $\rho_{\Pi'}(f(I),x) \geq 1 - c(\varepsilon)$ implique $\rho_\Pi(I,g(I,x)) \geq 1 - \varepsilon$. ■

On peut bien entendu définir de façon analogue la réduction **P** quand l'un des deux problèmes (ou les deux) est une minimisation. On obtient aisément la propriété suivante : *la réduction **P** préserve l'appartenance à **PTAS**.*

CRESCENZI et PANCONESI prouvent alors la complétude dans **APX** d'un problème dérivé de **MAX SAT PONDÉRÉ** appelé **MAX SAT PONDÉRÉ-B**. Dans ce problème, on nous donne une formule φ sur un ensemble de variables x_1, \dots, x_n , un poids w_i pour chaque variable x_i et un entier W . La différence avec **MAX SAT PONDÉRÉ** consiste en une hypothèse sur les poids, qui doivent vérifier $W \leq \sum_{i=1}^n w_i \leq 2W$. Comme précédemment, il s'agit de trouver une valeur de vérité satisfaisant φ et de poids maximal. Nous considérons ici, pour rentrer dans le cadre des problèmes de **NPO**, que la solution mettant toutes les variables à *faux* est réalisable et de valeur W . Ce problème est alors trivialement dans **APX**, puisque la solution consistant à mettre toutes les variables à *vrai* est une 2-approximation.

Théorème 6. ([19]) **MAX SAT PONDÉRÉ-B** est **APX-complet** sous la réduction **P**.

La preuve de ce théorème se base, comme les preuves de **NPO**-complétudes vues précédemment, sur la transformation d'une machine de **TURING** en formule **SAT**, donc sur la preuve du théorème de **COOK**. Ce résultat revêt une grande importance puisqu'il constitue le premier résultat de complétude dans une classe d'approximation, qui plus est la plus fameuse. Néanmoins, il soulève principalement deux questions.

La première est que le problème prouvé complet est « artificiel », il a été construit *ad hoc*. Il semble alors particulièrement intéressant de prouver la **APX**-complétude de problèmes naturels. La deuxième est que **MAX SAT PONDÉRÉ-B** n'admet de toute évidence pas de schéma (l'approcher à rapport meilleur que 2 permettrait de résoudre **SAT**), si **P** \neq **NP**. La proposition 2 (section 4.3) liant complétude et inapproximabilité n'est alors d'aucune utilité relativement aux résultats négatifs pour ce problème. Nous verrons cependant que si une réponse pourra être apportée à la première remarque, la deuxième apparaît récurrente, soulignant la primauté de l'aspect structurel vis-à-vis de l'aspect opérationnel de la notion de complétude.

Devant la difficulté de trouver d'autres problèmes, naturels si possible, **APX**-complets sous la réduction **P**, survient la question de savoir si cette réduction ne comporte pas intrinsèquement des éléments rendant ardue l'obtention de tels résultats. Dans cet ordre d'idée, CRESCENZI et TREVISAN ont montré dans [22] qu'elle n'était pas adaptée pour prouver des résultats de complétude à propos de problèmes polynomialement bornés. Le problème réside dans le fait que la réduction **P** transforme des solutions optimales en solutions optimales, au même titre que beaucoup d'autres réductions d'ailleurs (la réduction stricte, comme nous l'avons déjà mentionné, mais aussi la réduction **L**, la réduction **E** (voir section 7.1), ...). Notons \mathbf{P}^{SAT} et $\mathbf{P}^{\text{SAT}[O(\log(n))]}$ la classe des problèmes de décision résolubles en utilisant respectivement un nombre polynomial ou logarithmique d'appels à un oracle résolvant **SAT**. CRESCENZI et TREVISAN montrent que prouver la complétude dans **APX** d'un problème polynomialement borné sous une des réductions précitées (préservant l'optimalité des solutions) reviendrait à prouver que $\mathbf{P}^{\text{SAT}} = \mathbf{P}^{\text{SAT}[O(\log(n))]}$, ce qui paraît improbable ([22]). Pour contourner cette difficulté, les auteurs de [22] introduit la réduction suivante, notée **PTAS**-réduction.

Définition 8 ([22]) Soient Π et Π' deux problèmes de maximisation de **NPO**. On dit que Π se réduit à Π' sous la réduction **PTAS** s'il existe trois fonctions f , g et c telles que :

- pour tout $I \in \mathcal{I}_{\Pi}$ et tout $\varepsilon \in]0,1[$, $f(I,\varepsilon) \in \mathcal{I}_{\Pi'}$; le temps de calcul de f est polynomial en fonction de $|I|$;
- pour tout $I \in \mathcal{I}_{\Pi}$, tout $\varepsilon \in]0,1[$ et tout $x \in \text{sol}(f(I,\varepsilon))$, $g(I,\varepsilon,x) \in \text{sol}(\mathcal{I}_{\Pi})$; le temps de calcul de g est polynomial en fonction de $|I|$ et de $|x|$;
- $c :]0,1[\rightarrow]0,1[$;
- pour tout $I \in \mathcal{I}_{\Pi}$, tout $\varepsilon \in]0,1[$ et tout $x \in \text{sol}(f(I,\varepsilon))$: $\rho_{\Pi'}(f(I,\varepsilon),x) \geq 1 - c(\varepsilon)$ implique $\rho_{\Pi}(I,g(I,\varepsilon,x)) \geq 1 - \varepsilon$. ■

Là encore, on peut sans peine définir de façon analogue la réduction **PTAS** lorsqu'elle implique un (ou deux) problème(s) de minimisation. Nous pouvons immédiatement observer que la réduction **PTAS** est une généralisation de la réduction **P** et qu'elle en conserve la principale propriété, à savoir la préservation de l'appartenance à **PTAS**. Bien évidemment, la réduction **PTAS** préserve l'appartenance à **PTAS**.

L'unique différence entre la réduction **P** et la réduction **PTAS** est la possibilité dans cette dernière de faire dépendre les fonctions f et g de ε . Cette différence est

notable puisque l'on peut s'apercevoir que la réduction PTAS ne transforme pas nécessairement des solutions optimales en solutions optimales. Ainsi, sous cette réduction il nous est autorisé d'espérer obtenir des résultats de complétude dans la classe **APX** pour des problèmes polynomialement bornés. C'est ce qu'ont fait CRESCENZI et TREVISAN pour le problème MAX SAT PONDÉRÉ-PB. Ce problème est dérivé de MAX SAT PONDÉRÉ-B : les instances et les solutions réalisables sont les mêmes, on a simplement modifié la fonction objectif de manière à la rendre polynomialement bornée. Ainsi, si I est une instance sur n variables et x une valeur de vérité :

$$m(I,x) = n + \left\lfloor \frac{n(m'(I,x) - W)}{W} \right\rfloor$$

où m' est la valeur de la solution x sur I par rapport à MAX SAT PONDÉRÉ-B. La valeur de toute solution est alors majorée par $2n$. Toute solution réalisable étant de valeur au moins n , le problème est de toute évidence dans **APX**.

Théorème 7. ([22]) MAX SAT PONDÉRÉ-B se **PTAS**-réduit à MAX SAT PONDÉRÉ-PB. Ainsi, MAX SAT PONDÉRÉ-PB est **APX**-complet sous la réduction **PTAS**.

On connaît maintenant un très grand nombre de problèmes naturels polynomialement bornés ou non qui sont **APX**-complets sous la réduction PTAS.

5.2.2 Problèmes intermédiaires

Après avoir mentionné des résultats de complétude concernant la classe **APX**, abordons maintenant ceux concernant la recherche de problèmes intermédiaires, renseignant sur la richesse structurelle issue de la classification liée aux réductions. Cette problématique a été abordée dès le premier article concernant l'étude de la complétude dans **APX** ([19]). CRESCENZI et PANCONESI montrent l'existence de problèmes intermédiaires dans **APX** (par rapport à la classe **PTAS**), sous la réduction **P** qu'ils ont introduite. Ce résultat est prouvé par une technique de diagonalisation inspirée de celle utilisée par LADNER pour prouver l'existence de problèmes intermédiaires dans **NP** ([47]). Malheureusement, là encore, le problème montré intermédiaire a été construit *ad hoc* et ne correspond à aucun problème réel.

CRESCENZI, KANN et TREVISAN ont ensuite poursuivi cette étude et ont obtenu des résultats probants. Ils ont plus précisément exhibé un problème intermédiaire (quoique sous l'hypothèse, différente de $\mathbf{P} \neq \mathbf{NP}$, que la hiérarchie polynomiale n'est pas finie) très largement étudié, le problème BIN PACKING (une instance de ce problème consiste en la donnée de n nombres x_1, \dots, x_n entre 0 et 1 ; il s'agit alors de répartir ces nombres dans des boîtes de capacité unitaire (la somme des nombres assignés à une boîte doit être inférieure ou égale à 1) en minimisant le nombre de boîtes utilisées). Il est bien connu que ce problème est dans **APX** mais n'admet pas de schéma d'approximation (si $\mathbf{P} \neq \mathbf{NP}$), voir par exemple [36].

Remarquons que puisque la réduction PTAS est une généralisation de la réduction **P**, l'existence même de problèmes **APX**-intermédiaires (naturels ou non) sous PTAS était intéressante. Remarquons également que, pour la même raison, ce résultat reste

bien entendu valable sous la réduction P . L'importance de ce résultat (et plus généralement des résultats concernant les problèmes intermédiaires de [20]) peut être mise en évidence si l'on repense à la situation dans \mathbf{NP} pour les problèmes de décision. En effet, à ce jour, nous connaissons certes des problèmes intermédiaires depuis [47] mais aucun problème intermédiaire *naturel*. Il n'apparaissait donc pas évident de pouvoir obtenir un tel résultat pour la structure des classes d'approximation.

5.2.3 Limites de l'approche

Les résultats présentés jusqu'ici dans cette section répondent à la problématique précédemment explicitée en définissant des réductions préservant certains types d'approximation qui structurent les classes d'approximation. Nous avons vu qu'un des intérêts de cette problématique était ses liens avec l'inapproximabilité des problèmes. Or, aucun des résultats de complétude mentionnés n'apporte de résultats nouveaux concernant cette inapproximabilité.

Comme nous l'avons déjà mentionné dans la section 3.3.3, ces résultats tant attendus sont issus de la caractérisation de \mathbf{NP} par le théorème PCP ([3]), théorème majeur (et d'une grande complexité). On peut alors, par une vision rétrospective, se demander si l'on aurait pu parvenir à des résultats équivalents par le biais de la complétude. Était-il raisonnable de vouloir prouver la complétude dans \mathbf{APX} d'un problème comme $\mathbf{MAX\ 3SAT}$ (prouvant ainsi qu'il n'admet pas de schéma)? Plus généralement, cette interrogation s'inscrit dans celle, très souvent posée, consistant à trouver une preuve de l'inapproximabilité de problèmes comme $\mathbf{MAX\ 3SAT}$ sans utiliser le théorème \mathbf{PCP} . Notons que très récemment, les auteurs de [28, 29] ont proposé une preuve combinatoire du théorème \mathbf{PCP} , donc de l'inapproximabilité de $\mathbf{MAX\ 3SAT}$.

CRESCENZI et TREVISAN ont étudié les rapports entre théorème PCP et complétude dans [21]. Leur angle d'attaque du problème a été d'analyser si d'une preuve de complétude de $\mathbf{MAX\ 3SAT}$ (dans \mathbf{APX} sous la réduction PTAS) on pouvait déduire le théorème \mathbf{PCP} . Il s'agit d'une certaine manière de mesurer la difficulté d'obtention d'une telle preuve de complétude. Ils répondent par l'affirmative en exhibant un lien fort entre complétude et théorème \mathbf{PCP} . Notons qu'une analyse réciproque (peut-on déduire du théorème \mathbf{PCP} des résultats de complétude?) sera d'ailleurs l'objet de la section 7. Détaillons brièvement le résultat de [21] en revenant quelques instants sur \mathbf{PCP} . Ce théorème stipule que $\mathbf{NP} = \mathbf{PCP}[\mathbf{O}(\log(n)), \mathbf{O}(1)]$. Autrement dit, cela ramène la question « $\mathbf{P} = \mathbf{NP}$? » à la question « $\mathbf{PCP}[\mathbf{O}(\log(n)), \mathbf{O}(1)] = \mathbf{P}$? ». CRESCENZI et TREVISAN prouvent alors que la complétude de $\mathbf{MAX\ 3SAT}$ entraînerait une version légèrement plus faible du théorème \mathbf{PCP} exprimée par le théorème suivant.

Théorème 8. ([21]) *Si $\mathbf{MAX\ 3SAT}$ est \mathbf{APX} -complet sous la réduction PTAS, alors*

$$\mathbf{PCP}[\mathbf{O}(\log(n)), \mathbf{O}(1)] = \mathbf{P} \implies \mathbf{NP} = \mathbf{co-NP}$$

La complétude de $\mathbf{MAX\ 3SAT}$ implique donc l'existence de problèmes non polynomiaux dans $\mathbf{PCP}[\mathbf{O}(\log(n)), \mathbf{O}(1)]$ sous l'hypothèse $\mathbf{NP} \neq \mathbf{co-NP}$ (au lieu de $\mathbf{P} \neq \mathbf{NP}$ pour le théorème \mathbf{PCP}).

En d'autres termes, comme expliqué dans [21], ce résultat signifie que toute preuve de **APX**-complétude de MAX 3SAT contient en elle-même une preuve (d'une version un peu moins forte) du théorème **PCP**. Cela explique donc pourquoi un tel résultat de complétude n'a pas été prouvé indépendamment.

5.3 PTAS-complétude

Nous présentons ici les résultats structurels concernant la classe des problèmes admettant un schéma d'approximation. Cette problématique a été abordée pour la première fois par CRESCENZI et PANCONESI dans [19].

La hiérarchie des classes d'approximation nous indique qu'une notion de **PTAS**-complétude serait intéressante surtout dans le cas d'une réduction préservant l'appartenance à **FPTAS**. Crescenzi et Panconesi ont ainsi défini la réduction F .

Définition 9. ([19]) Soient Π et Π' deux problèmes de **NPO**. On dit que Π se réduit à Π' sous la réduction F s'il existe trois fonctions f , g et c telles que :

- pour tout $I \in \mathcal{I}_\Pi$, $f(I) \in \mathcal{I}_{\Pi'}$; f est calculable en temps polynomial ;
- pour tout $I \in \mathcal{I}_\Pi$ et tout $x \in \text{sol}(f(I))$, $g(I,x) \in \text{sol}(\mathcal{I}_\Pi)$; g est calculable en temps polynomial ;
- la complexité du calcul de c est bornée par $p(1/\varepsilon, |x|)$, où p est un polynôme ;
- la valeur de c est $1/q(1/\varepsilon, |x|)$, où q est un polynôme ;
- pour tout $I \in \mathcal{I}_\Pi$ et tout $x \in \text{sol}(f(I))$: $|1 - \rho_{\Pi'}(f(I), x)| \leq c(1/\varepsilon, |x|)$ implique $|1 - \rho_\Pi(I, g(I, x, r))| \leq \varepsilon$. ■

Nous obtenons aisément la propriété suivante : *la réduction F préserve l'appartenance à **FPTAS***. Partant de cette réduction, CRESCENZI et Panconesi ont étudié la complétude dans **PTAS** et ont montré que le problème $\text{MAX SAT PONDÉRÉ-}B\text{-LINÉAIRE}$ est **PTAS**-complet. Ce problème est une variante du problème $\text{MAX SAT PONDÉRÉ-}B$, défini dans la section 5.2 : le seul changement réside dans le fait que la somme des poids doit être comprise entre W et $(1 + (1/n - 1))W$ (où n est le nombre de variables).

Ce problème est évidemment dans **PTAS** (par ailleurs, il est construit pour). Étant donnée une constante $\varepsilon > 0$, pour avoir un algorithme $(1 - \varepsilon)$ -approché, il suffit de renvoyer n'importe quelle solution si $n \geq 1/\varepsilon$ et de résoudre exactement par recherche exhaustive les cas où $n < 1/\varepsilon$.

Nous ne détaillerons pas la preuve de la complétude, basée sur la simulation d'une machine de Turing particulière et analogue à celle du théorème 6.

Théorème 9. ([19]) $\text{MAX SAT PONDÉRÉ-}B\text{-LINÉAIRE}$ est **PTAS**-complet sous la réduction F .

Il est à noter que ce problème est l'unique problème à avoir été montré **PTAS**-complet sous la réduction F . En particulier, nous ne connaissons pas de problèmes naturels **PTAS**-complets (sous F).

De la même manière que pour la **APX**-complétude, CRESCENZI et PANCONESI ont ensuite raffiné leur étude en montrant l'existence de problèmes qui ne sont ni **PTAS**-complets ni dans **FPTAS** (si $\mathbf{P} \neq \mathbf{NP}$), c'est-à-dire de problèmes intermédiaires.

Comme nous l'avons déjà remarqué, **MAX SAT PONDÉRÉ- B -LINÉAIRE**, le seul problème ayant pu être démontré **FPTAS**-complet sous la réduction F est un problème très artificiel. Par ailleurs, il semble que la réduction F elle-même soit très contraignante pour permettre l'existence d'autres problèmes **FPTAS**-complets. Plus généralement, on aurait pu se demander si la démarche basée sur la conception de réductions (préservant l'approximabilité) calquées sur le modèle des réduction de **KARP** (comme le sont toutes les réductions vues jusqu'à maintenant dans cet article) n'est pas intrinsèquement limitée surtout quand il s'agit des classes « proches » de **P** (comme la **FPTAS**). Dans ce cas, on aurait besoin d'un modèle de réduction de base plus puissant pour inspirer des réduction préservant l'approximabilité pour ce type de classes d'approximabilité. Dans [13], on se base justement sur le modèle de la réduction de **TURING** présentée dans la section 2. En effet, nous allons supposer l'existence d'un oracle résolvant le problème d'arrivée de manière approchée, et nous réduirons le problème de départ au problème d'arrivée en exhibant un algorithme approché utilisant cet oracle. C'est donc cette fois-ci l'adaptation d'une réduction de Turing au cadre de l'optimisation. On arrive ainsi la réduction suivante appelée FT .

Définition 10. ([13]) Soient Π et Π' deux problèmes d'optimisation appartenant à **NPO**. Soit de plus un oracle $\bigcirc_{\alpha}^{\Pi'}$ fournissant, pour toute instance I' de Π' (et pour tout $\alpha > 0$), une solution réalisable x de I' $(1 - \alpha)$ -approchée, si $\text{but}(\Pi') = \max$, $(1 + \alpha)$ -approchée sinon. On dit que Π se réduit à Π' sous la réduction FT si l'on dispose, pour tout $\varepsilon > 0$, d'un algorithme $A_{\varepsilon}(I, \bigcirc_{\alpha}^{\Pi'})$ tel que :

- pour toute instance I de Π , A_{ε} fournit une solution réalisable x $(1 - \varepsilon)$ -approchée $\text{but}(\Pi) = \max$, $(1 + \varepsilon)$ -approchée sinon ;
- si nous supposons que, pour toute instance I' de Π' , l'oracle $\bigcirc_{\alpha}^{\Pi'}(I')$ fournit une solution de I' $(1 - \alpha)$ -approchée en temps polynomial en $|I'|$ et en $1/\alpha$, alors le temps d'exécution de A_{ε} est polynomial en fonction de la taille de l'instance et de $1/\varepsilon$. ■

Il est aisé de voir que la réduction FT ainsi définie préserve l'existence de schémas complets, *i.e.*, elle préserve l'appartenance à **FPTAS**.

Théorème 10. ([13]) Soit un problème $\Pi \in \mathbf{NPO}$ dont la version décision est **NP-complète**. Si Π est polynomialement borné, alors, pour tout problème $\Pi' \in \mathbf{NPO}$, $\Pi' \leq_{FT} \Pi$. Par conséquent, si un problème de **NPO** est dans **PTAS**, polynomialement borné et **NP-complet** dans sa version décision, alors il est **PTAS-complet** sous la réduction FT .

Les deux problèmes classiques **MAX STABLE PLANAIRE** et **MIN COUVERTURE DE SOMMETS PLANAIRE** admettent un schéma ([12]) et sont polynomialement bornés. Ainsi, le théorème suivant est un corollaire immédiat du théorème 10.

Théorème 11. ([13]) Les problèmes **MAX STABLE PLANAIRE** et **MIN COUVERTURE DE SOMMETS PLANAIRE** sont **PTAS-complets** sous la réduction FT .

La réduction FT est moins restrictive que la réduction F qui en constitue un cas particulier. Cela transparait d'ailleurs dans le théorème 10 où l'on montre que l'on peut,

sous des hypothèses relativement faibles, réduire tout problème de **NPO** à un problème de **PTAS**. Ce résultat peut s'exprimer en termes de fermeture : *la fermeture de PTAS sous FT est égale à NPO*. Vient alors évidemment la question de l'existence de problèmes intermédiaires. Plus une réduction est large, moins il peut y avoir de problèmes intermédiaires. Les réductions que nous avons proposées sont-elles si peu contraignantes que dès qu'un problème n'est pas dans **FPTAS** il est **PTAS**-complet? Les préordres induits par ces réductions n'ont-ils que deux niveaux?

La principale difficulté pour étudier cette question tient à la nature même de la réduction FT. En effet, comme nous l'avons déjà dit, elle s'apparente à des réductions de TURING et non à des réductions de KARP. Il apparaît donc plus difficile d'utiliser par exemple une technique de diagonalisation comme celle proposée par LADNER pour obtenir des problèmes intermédiaires dans **NP**. D'ailleurs, comme nous l'avons déjà mentionné dans la section 2, l'existence même de problèmes intermédiaires dans **NP** (sous l'hypothèse $P \neq NP$) sous la réduction de TURING est à notre connaissance une question ouverte. Il semble alors délicat de trouver des résultats intermédiaires sous la réduction FT. Face à ce problème, nous pouvons étudier les problèmes intermédiaires sous une hypothèse autre que $P \neq NP$. Nous cherchons ainsi à conditionner l'existence de problèmes **PTAS**-intermédiaires à celle de problèmes intermédiaires sous une réduction de TURING. La version optimisation de la réduction de TURING, préserve l'appartenance à **PO**. Il est démontré dans [13] que l'existence de problèmes **NPO**-intermédiaires (relativement à **PO**) sous la réduction de TURING adaptée aux problèmes d'optimisation est une condition suffisante à l'existence de problèmes **PTAS**-intermédiaires.

Théorème 12. ([13]) *S'il existe un problème de NPO intermédiaire sous la réduction de TURING (relativement à PO), alors il existe un problème PTAS-intermédiaire sous FT.*

6 Max-SNP, Max-NP et la réduction L

Nous avons vu dans la section 3.4 des liens entre logique, complexité et approximation, apparaissant notamment dans la définition des classes syntaxiques **Max-SNP** et **Max-NP** dans [55]. Ces deux classes sont des sous-classes de **APX**, et contiennent un grand nombre de problèmes classiques. S'attaquant par l'approche structurelle au problème de l'existence de schémas pour ces problèmes, PAPADIMITRIOU et YANNAKAKIS cherchent alors à tirer parti de la définition syntaxique de leurs classes pour chercher à obtenir des résultats de complétude.

La première étape consiste à définir une réduction préservant l'existence des schémas d'approximation ; c'est ce qui a été fait avec la réduction L, que nous avons déjà présentée (voir section 4.1). L'avantage des classes **Max-SNP** et **Max-NP** tient au fait que les problèmes sont définis de manière plus structurée que des problèmes de **NPO** quelconques. C'est grâce à cette structure que PAPADIMITRIOU et YANNAKAKIS ont élaboré une réduction générique vers **MAX 3SAT**.

Théorème 13. ([55]) *MAX 3SAT est Max-SNP-complet sous la réduction L.*

Ce résultat est fondamental. En effet, il ramène le problème de l'existence de schéma pour $\text{MAX } 3\text{SAT}$ à celle d'un schéma pour tous les problèmes de **Max-SNP**. Cette classe contenant un grand nombre de problèmes naturels, ce résultat, s'il ne résout pas directement la question, permet néanmoins de pencher vers l'absence de schéma, et s'inscrit dans un esprit similaire à celui de la **NP**-complétude de **SAT** (ramener une question sur un problème à la même question pour tout un ensemble de problèmes).

Corollaire 3. ([55]) $\text{MAX } 3\text{SAT}$ est dans **PTAS** si et seulement si **Max-SNP** est inclus dans **PTAS**.

Notons que ce résultat de complétude se démarque des précédents (de la section 5) de part la manière dont il a été obtenu. Il n'est nullement basé sur la simulation d'une machine de **TURING**, sur la preuve du théorème de **COOK**. Sa démonstration est basée sur des transformations de formules logiques, ce qui a été rendu possible bien évidemment grâce à la définition logique des problèmes de **Max-SNP**.

PAPADIMITRIOU et **YANNAKAKIS** se sont ensuite attachés à prouver la **Max-SNP**-complétude d'autres problèmes classiques, par réduction depuis $\text{MAX } 3\text{SAT}$. Ils ont ainsi prouvé la **Max-SNP**-complétude de nombreux problèmes, comme par exemple $\text{MAX STABLE-}B$ ou MAX COUPE , indiquant donc que l'existence de schéma pour chacun d'entre eux était équivalente à l'existence de schéma pour tout problème de **Max-SNP**. Si des résultats de complétude ont donc été obtenus pour **Max-SNP**, il n'en va pas de même pour **Max-NP**. Il aurait été intéressant de montrer la **Max-NP**-complétude de MAX SAT sous la réduction **L** en utilisant par exemple une technique de même nature que celle adoptée pour prouver le théorème 13. Le questionnement, soulevé par [55], est en fait double. D'une part, trouver, comme nous l'avons dit, des problèmes complets, naturels si possible, pour **Max-NP**. D'autre part, étudier les rapports entre **Max-SNP** et **Max-NP**. On sait que d'un côté l'inclusion est stricte, mais que d'un autre les problèmes de **Max-NP** sont approximables à rapport constant. Ce dernier point laisse à penser que si la structure d'un problème de **Max-NP** est potentiellement plus « riche », il n'est pas forcément plus difficile à résoudre de manière approchée qu'un problème de **Max-SNP**. D'où la question : peut-on réduire tout problème de **Max-NP** à un problème de **Max-SNP**? Remarquons que répondre à cette question permet de répondre immédiatement à la précédente.

C'est à ces deux questions qu'ont répondu **CRESCENZI** et **TREVISAN** dans [21]. Se heurtant au caractère relativement strict de la réduction **L**, ils étudient la complétude dans **Max-NP** sous la réduction **PTAS**, plus souple. Ils parviennent ainsi à montrer que le problème $\text{MAX } 3\text{SAT}$ est **Max-NP-complet** sous la réduction **PTAS**. Remarquons que, nous le verrons en détail dans la section 7, ce résultat était déjà connu. En effet, grâce au théorème **PCP**, on a pu montrer que le problème $\text{MAX } 3\text{SAT}$ était **APX**-complet sous la réduction **PTAS** (voir section 7.2). **Max-NP** étant inclus dans **APX**, la **Max-NP**-complétude de $\text{MAX } 3\text{SAT}$ n'est donc qu'un corollaire d'un résultat plus général.

Tout l'intérêt du résultat de [21] réside dans sa preuve qui *ne fait pas usage du théorème PCP*. La démonstration utilise en effet une réduction générique directe, transformant structurellement les problèmes de **Max-NP** en problèmes de **Max-SNP**. Cette

réduction a notamment l'avantage d'être constructive. L'instance de $\text{MAX } 3\text{SAT}$ est effectivement construite (contrairement à la preuve utilisant le théorème **PCP**), et l'on peut ainsi comprendre comment transformer par exemple MAX SAT en $\text{MAX } 3\text{SAT}$ en préservant une certaine exigence quant à l'approximation.

Notons enfin que MAX SAT ainsi que tous les problèmes montrés **Max-SNP**-complets sous la réduction L , sont **Max-NP**-complets sous la réduction PTAS .

7 Complétude en utilisant PCP

Si l'on résume (très) rapidement les résultats exposés dans les sections 5 et 6, nous pourrions les opposer de la manière suivante :

- en utilisant des preuves s'inspirant de la preuve du théorème de **COOK** et en utilisant des réductions basées sur la réduction de **KARP**, on obtient des problèmes complets pour des classes d'approximation (**APX** et **PTAS**) mais ces problèmes ne sont pas naturels ;
- en définissant des classes syntaxiques, on obtient des problèmes complets pour une sous-classe de **APX** ; cette démarche, nous l'avons mentionné auparavant, s'inscrit dans le même ordre d'idée que la **NP**-complétude, mais ne permet apparemment pas d'obtenir des résultats sur la classe **APX** en entier.

Le théorème **PCP** a permis de répondre (par la négative) à la question de l'approximabilité d'un nombre important de problèmes classiques. Parmi ceux-là, le premier est $\text{MAX } 3\text{SAT}$, problème central dans **Max-SNP**. L'idée qui vient à l'esprit est alors d'utiliser la puissance de ce théorème pour prouver des résultats de complétude. Peut-on, grâce au système **PCP**, ou plutôt aux systèmes **PCP**, établir de nouvelles réductions génériques ?

C'est la tâche à laquelle se sont attelés **KHANNA**, **MOTWANI**, **SUDAN** et **VAZIRANI**. Ils sont parvenus à des conclusions très satisfaisantes, parues dans [43]. Ils ont non seulement obtenu des résultats de complétude pour des problèmes naturels, ce qui est en soi d'une importance majeure, mais leur étude va plus loin. Leurs résultats de complétude se déduisent d'une méthode générale liant complétude et inapproximabilité. Un résultat d'inapproximabilité, obtenu par exemple par **PCP**, se transforme alors grâce à leur méthode en un résultat de complétude. Cette méthode générale s'applique en outre à différentes classes d'approximation. Un résultat de complétude pour une classe d'approximation peut se voir simplement comme une instanciation particulière de la méthode pour la classe (et le problème) dont il est question. De tels résultats complètent donc ceux obtenus par les deux approches précédemment exposées, et apportent notamment une réponse à la question centrale de la complétude de $\text{MAX } 3\text{SAT}$ pour la classe **APX**.

7.1 De PCP à la complétude

L'idée originelle de **KHANNA** *et al.* ([43]) semble avoir été d'utiliser la réduction L . Ils ont cependant été amenés à définir une réduction légèrement plus large, la rédu-

tion E.

Définition 11. ([43]) Soient deux problèmes de minimisation Π et Π' de **NPO**. On dit que Π se réduit à Π' sous la réduction E s'il existe deux fonctions f et g calculables en temps polynomial, une constante β et un polynôme p tels que :

- pour tout $I \in \mathcal{I}_\Pi$, $f(I) \in \mathcal{I}_{\Pi'}$. De plus, $\text{opt}_{\Pi'}(f(I)) \leq p(|I|) \text{opt}_\Pi(I)$;
- pour tout $I \in \mathcal{I}_\Pi$ et tout $x \in \text{sol}(f(I))$, $g(I,x) \in \text{sol}(\mathcal{I}_\Pi)$; de plus : $\rho_\Pi(I, g(I,x)) - 1 \leq \beta(\rho_{\Pi'}(f(I), x) - 1)$. ■

La réduction E peut se définir de manière analogue si l'un des deux problèmes (ou les deux) consistent en une maximisation. Remarquons que la propriété requise sur la transformation des rapports d'approximation est semblable à celle induite par la réduction L (voir la remarque suivant la définition 3 de la réduction L). La différence entre ces deux réductions consiste simplement à relâcher la dépendance linéaire imposée par la réduction L entre les valeurs optimales en une dépendance polynomiale. La réduction E conserve ainsi la propriété principale de la réduction L, à savoir la préservation de l'appartenance à **PTAS**.

Abordons maintenant la relation entre inapproximabilité et complétude. Il s'agit de remarquer dans un premier temps que les résultats fournis par **PCP** peuvent se réécrire sous une forme particulière (appelée difficulté canonique).

Introduisons d'abord les définitions suivantes ([43]). Une famille \mathcal{F} de fonctions de \mathbb{N} dans \mathbb{N} est dite *close inférieurement* si pour tout $g \in \mathcal{F}$ et pour toute constante c , si $h(n) = O(g(n^c))$ alors $h \in \mathcal{F}$. Une fonction g est dite *difficile* pour \mathcal{F} si pour tout $h \in \mathcal{F}$ il existe une constante c telle que $h(n) = O(g(n^c))$. La fonction g est dite *complète* si de plus $g \in \mathcal{F}$. Enfin, **\mathcal{F} -APX-PB** désigne les problèmes polynomialement bornés approximables à rapport $g(n)$ pour un certain $g \in \mathcal{F}$.

Nous pouvons d'ores et déjà remarquer que la classe des fonctions majorées par une constante, celle des fonctions majorées par une fonction logarithmique et celle des fonctions majorées par un polynôme sont closes inférieurement. Les trois classes de problèmes correspondantes sont les trois classes déjà définies. Notons également que les fonctions constantes, logarithmiques et polynomiales sont (respectivement) complètes pour ces trois classes de fonctions.

Par ailleurs, un problème de maximisation de **NPO** Π est dit *canoniquement difficile* pour la classe **\mathcal{F} -APX**, où \mathcal{F} est une famille de fonctions close inférieurement, s'il existe une fonction f calculable en temps polynomial, des constantes n_0 et c , et une fonction G difficile pour \mathcal{F} telles que :

1. pour toute instance φ de 3SAT sur n variables, et pour tout $N \geq n^c$, $f(\varphi, N)$ est une instance de Π ;
2. si φ est satisfiable, alors $\text{opt}(f(\varphi, N)) = N$;
3. si φ n'est pas satisfiable, alors $\text{opt}(f(\varphi, N)) = N/G(N)$;
4. étant donnée une solution x de $f(\varphi, N)$ de valeur supérieure à $N/G(N)$ (l'inégalité est stricte), on peut retrouver en temps polynomial une valeur de vérité satisfaisant φ .

Comme indiqué dans [43], nous définissons de manière analogue les problèmes de minimisation canoniquement difficiles (en remplaçant dans le deuxième item $N/G(N)$ par $NG(N)$ et en modifiant les items 3 et 4 en conséquence).

Remarque 1. Grâce à la **NP**-complétude de $3SAT$, nous obtenons une définition équivalente en disant que le problème est canoniquement difficile si l'on peut lui réduire tout problème de **NP** selon les items 1 à 4 (en les adaptant bien sûr convenablement). Suivant cette remarque, la notion de difficulté canonique apparaît très reliée à la réduction **GAP**. Elle consiste en effet en une sorte de réduction **GAP** valable pour une famille de fonctions, à ceci près cependant que l'on considère ici que l'on peut récupérer un certificat prouvant que l'instance est positive lorsque c'est la cas (dernier item de la définition). Cette légère différence est cependant indispensable dans la construction proposée par [43]. ■

L'intérêt de cette définition des problèmes canoniquement difficiles est qu'elle correspond aux résultats fournis par le théorème **PCP**.

Ces résultats nous fournissent donc une famille de réductions de type (un peu plus puissant que) **GAP**. Cependant, les notions de complétude requièrent des réductions entre problèmes d'optimisation. Tout le problème consiste alors à utiliser le pouvoir des réductions fournies par les résultats issus du théorème **PCP** pour obtenir des réductions d'un problème d'optimisation à un autre problème d'optimisation. KHANNA *et al.* ont résolu cette difficulté en concevant une réduction d'un problème d'optimisation Π à un problème d'optimisation Π' (canoniquement difficile pour une certaine classe de problèmes) en utilisant les réductions à Π' de tout un ensemble de problèmes de décision liés à Π .

Avant de détailler l'idée de cette méthode, nous devons introduire une dernière notion, celle de problème additif. Un problème Π de **NPO** est dit *additif* s'il existe deux fonctions \oplus et f calculables en temps polynomial telles que :

- pour tout couple (I_1, I_2) d'instances de Π , $I_1 \oplus I_2$ est une instance de Π telle que $\text{opt}(I_1 \oplus I_2) = \text{opt}(I_1) + \text{opt}(I_2)$;
- pour toute solution x de $I_1 \oplus I_2$, $f(x)$ est un couple (x_1, x_2) de solutions pour I_1 et I_2 , respectivement, tel que $m_{I_1 \oplus I_2}(x) = m_{I_1}(x_1) + m_{I_2}(x_2)$.

La relation entre la difficulté canonique et la complétude dans les classes d'approximation est étroite. Elle est présentée dans le théorème suivant, qui constitue la pierre angulaire permettant de faire le lien entre inapproximabilité via le théorème **PCP** et complétude dans les classes d'approximation.

Théorème 14. ([43]) *Si \mathcal{F} est une famille close inférieurement, alors tout problème additif Π canoniquement difficile pour la classe \mathcal{F} -APX-PB est \mathcal{F} -APX-PB-difficile sous la réduction E .*

7.2 APX-complétude

La première application du théorème **PCP** pour l'inapproximabilité des problèmes d'optimisation fournit un résultat négatif pour **MAX 3SAT**. Partant d'une instance I

d'un problème Π de **NP**, **PCP** permet de construire une instance φ de **MAX 3SAT** telle que ([4, 43]) :

- si I est positive alors φ est satisfiable ;
- si I est négative alors au plus une fraction $(1 - \varepsilon)$ des clauses de φ sont satisfiables ;
- si une solution satisfait strictement plus qu'une fraction $(1 - \varepsilon)$ des clauses de φ , on peut retrouver en temps polynomial un certificat prouvant que I est positive.

Théorème 15. ([4]) *Le problème **MAX 3SAT** est canoniquement difficile pour **APX-PB**.*

En combinant les théorèmes 14 et 15, nous aboutissons au résultat fondamental suivant.

Théorème 16. ([43]) *Le problème **MAX 3SAT** est **APX-PB-complet** sous la réduction E .*

La connexion est ainsi effectuée, nous aboutissons au premier résultat de complétude via **PCP**. Il établit la complétude de l'un des problèmes les plus classiques pour une classe qui ne l'est pas moins. On pourrait objecter que l'on aimerait avoir un résultat de complétude pour la classe **APX** en entier. Comme l'ont fait remarquer dans [43], il suffit pour cela de considérer le réduction **PTAS** au lieu de la réduction E .

Théorème 17. ([43]) *Le problème **MAX 3SAT** est **APX-complet** sous la réduction **PTAS**.*

Ayant obtenu la complétude dans **APX** de **MAX 3SAT**, nous pouvons immédiatement en déduire celle de nombreux autres problèmes. En particulier, dans [55] il est montré, par des réductions L partant de **MAX 3SAT**, que plusieurs autres problèmes étaient **Max-SNP-complets**. De tels problèmes (comme **MAX STABLE- B** ou **MAX COUPE**) sont donc **APX-complets** (sous la réduction **PTAS**). Depuis, ces problèmes classiques ont servi de base à l'élaboration de réductions (L , E , ou plus généralement **PTAS**) vers un très grand nombre de problèmes d'optimisation. La classe des problèmes **APX-complets** contient effectivement une quantité énorme de problèmes. Nous invitons une nouvelle fois le lecteur à consulter le compendium dans [7].

7.3 Complétude dans Poly-APX

7.3.1 Complétude dans Poly-APX-PB et Log-APX-PB

Comme expliqué précédemment, une des qualités majeures du travail de [43] est d'avoir proposé un lien générique entre inapproximabilité et complétude, lien qui peut s'appliquer à plusieurs classes d'approximation. Nous allons présenter les autres résultats de complétude obtenus dans [43], qui concernent deux classes correspondant à deux familles closes inférieurement : celle des fonctions majorées par un polynôme et celle des fonctions majorées par une fonction logarithmique.

Des liens ont été établis entre **MAX STABLE** et les systèmes de preuves interactives dès les premières apparitions de ceux-ci, permettant d'obtenir un résultat d'inapproximabilité de n^c , pour un certain $c > 0$ ([35]). Ces résultats ont ensuite été améliorés, par une optimisation des systèmes **PCP**, pour obtenir ce résultat avec $c = 1/2$ ([39]).

Concernant la problématique qui nous intéresse, nous pouvons formuler ces résultats comme suit.

Théorème 18. ([35, 39]) *Le problème MAX STABLE est canoniquement difficile pour la classe **Poly-APX-PB**.*

En combinant alors les théorèmes 14 et 18, nous obtenons le théorème suivant.

Théorème 19. ([43]) *Les problèmes MAX STABLE et MAX CLIQUE sont **Poly-APX-PB**-complets sous la réduction E.*

Des développements ultérieurs aux travaux de [43] ont abouti à un résultat d'inapproximabilité concernant le problème MIN SET COVER, bien connu pour être approximable à rapport $1 + \ln(n)$ (où n est le nombre d'éléments à couvrir) depuis l'article fondateur de JOHNSON ([41]). Selon RAZ et SAFRA ([58]), MIN SET COVER est difficile à approcher avec un rapport $c \ln(n)$ pour une constante $c > 0$. Avec la même démarche que précédemment nous obtenons le résultat suivant.

Théorème 20. ([58, 43]) *Le problème MIN SET COVER est canoniquement difficile pour la classe **Log-APX-PB**. Par conséquent, MIN SET COVER est **Log-APX-PB**-complet sous la réduction E.*

7.3.2 Complétude dans la classe **Poly-APX**

Malheureusement, si nous voulons généraliser ce résultat à toute la classe **Poly-APX**, nous nous heurtons à une difficulté majeure. Reprenons un instant le cas de la complétude dans **APX**. Nous avons mentionné dans la section 5.2 qu'il était peu probable que l'on puisse prouver la complétude dans **APX** d'un problème polynomialement borné sous une réduction préservant l'optimalité des solutions, puisque ceci montrerait $\mathbf{P}^{\text{SAT}} = \mathbf{P}^{\text{SAT}[O(\log(n))]}$. Ceci avait motivé l'introduction de réductions ne préservant pas l'optimalité des solutions, telle la réduction PTAS. Partant de ce constat, il semble donc vain de vouloir obtenir un résultat de complétude pour MAX STABLE ou MAX CLIQUE pour la classe **Poly-APX** si nous conservons la réduction E.

Dans [13], nous nous intéressons à étudier la possibilité de démontrer la complétude dans **Poly-APX** en utilisant non la réduction E mais plutôt la moins restrictive réduction PTAS. Le parallèle entre difficulté canonique et complétude s'exprime alors sous la forme suivante ([13]).

Théorème 21. ([13]) *Si Π' est un problème de maximisation additif canoniquement difficile pour **Poly-APX**, alors tout problème de maximisation de **Poly-APX** se PTAS-réduit à Π' .*

Notons que si Π est un problème de minimisation, on peut le PTAS-réduire à un problème de maximisation de **Poly-APX**, comme cela est indiqué dans [43].

Nous avons donc un résultat de complétude pour toute la classe **Poly-APX**. En réutilisant alors les résultats de difficulté canonique de **MAX STABLE** et **MAX CLIQUE**, nous obtenons immédiatement le résultat suivant.

Théorème 22. ([13]) *MAX STABLE et MAX CLIQUE sont **Poly-APX**-complets sous la réduction PTAS.*

7.3.3 Complétude dans la classe **Log-APX**

L'observation faite pour **Log-APX** au début du paragraphe 7.3.2 reste valide aussi pour la classe **Log-APX**. Donc, si l'on veut étudier l'existence de problèmes complets pour la classe **Log-APX** toute entière, on doit le faire en utilisant des réductions plus « larges » que la E-réduction.

Dans [32], une légère modification de la réduction PTAS, la réduction MPTAS est introduite. Elle est définie comme suit.

Définition 12. ([32]) Soient Π and Π' deux problèmes de maximisation de **NPO** (le cas de minimisation est complètement analogue). On dit que, Π se réduit à Π' sous la réduction MPTAS, si et seulement si il existe deux fonctions f et g , calculables en temps polynomial, et une fonction c telles que :

- pour tout $I \in \mathcal{I}_\Pi$ et tout $\varepsilon \in]0,1[$, $f(I,\varepsilon) = (I'_1, I'_2, \dots, I'_M)$ est une famille d'instances de Π' (où M est polynomialement borné en $|I|$);
- pour tout $I \in \mathcal{I}_\Pi$, tout $\varepsilon \in]0,1[$ et toute famille de solutions réalisables $x = (x_1, x_2, \dots, x_M)$, où x_i est une solution réalisable de I'_i , $g(I, x, \varepsilon) \in \text{sol}(I)$;
- $c :]0,1[\rightarrow]0,1[$;
- il existe un indice j tel que, quel que soit $I \in \mathcal{I}_\Pi$ et $\varepsilon \in]0,1[$: $\rho_{\Pi'}(I'_j, x_j) \geq 1 - c(\varepsilon)$ implique $\rho_\Pi(I, g(I, x, \varepsilon)) \geq 1 - \varepsilon$. ■

Il est facile de voir que la réduction MPTAS préserve l'appartenance à **PTAS**. Aussi, le fait que la fonction f dans la définition 12 soit multivaluée relâche la restriction aux problèmes additifs et s'applique même à ceux qui ne le sont pas.

En utilisant la réduction MPTAS, le théorème suivant peut être démontré ([32]).

Théorème 23. ([32]) *Soit \mathcal{F} une classe de fonctions closes inférieurement, et $\Omega \in \mathbf{NPO}$ un problème de maximisation canoniquement difficile pour la classe $\mathcal{F} - \mathbf{APX}$. Alors tout problème de maximisation de $\mathbf{NPO} \cap \mathcal{F} - \mathbf{APX}$ se réduit à Ω sous la réduction MPTAS.*

On peut montrer (cf. [43]) que tout problème de minimisation de $\mathcal{F} - \mathbf{APX}$ se E-réduit (donc, se MPTAS-réduit aussi) à un problème de maximisation de $\mathcal{F} - \mathbf{APX}$ ce qui nous conduit à la généralisation suivante du théorème 23.

Théorème 24. ([32]) *Soit \mathcal{F} une classe de fonctions closes inférieurement, et $\Omega \in \mathbf{NPO}$ un problème canoniquement difficile pour la classe $\mathcal{F} - \mathbf{APX}$. Alors tout problème de $\mathbf{NPO} \cap \mathcal{F} - \mathbf{APX}$ se réduit à Ω sous la réduction MPTAS.*

Considérons maintenant la classe **Log-APX**. Le problème **MIN SET COVER**, une instance duquel est définie par un système d'ensembles \mathcal{S} sur un ensemble de base C , est approximable à rapport $O(\log |C|)$ ([59]) ; par conséquent, il appartient à **Log-APX**. Il est de plus inapproximable à rapport $c \log |C|$, pour une constante c , à moins que $\mathbf{P} = \mathbf{NP}$ (cf. [5, 58] où ce résultat est mentionné et aussi [32] pour une démonstration informelle). Donc, en appliquant le théorème 24, le résultat suivant peut être dérivé pour **MIN SET COVER**.

Théorème 25. ([32]) **MIN SET COVER** est **Log-APX-complet** sous la **MPTAS-réduction**.

8 Complétude en approximation différentielle

L'étude de l'approximation selon le rapport différentiel s'est développée, nous l'avons mentionné en introduction, principalement depuis le début des années 90. Après l'explicitation d'arguments opérationnels et mathématiques justifiant l'utilisation du rapport ([27]), a commencé une étude systématique des problèmes d'optimisation (selon l'approximation différentielle). Des résultats (positifs et négatifs) sont ainsi apparus pour des problèmes classiques (citons notamment la coloration ([23, 30, 38]), le voyageur de commerce ([51, 50, 14, 37]), **BIN PACKING** ([24, 25])); des aspects structurels et opérationnels sont aussi abordés dans [52], etc. C'est tout naturellement que, comme dans le cadre de l'approximation standard, la problématique de la structure dans les classes d'approximation a ensuite été abordée. Si des notions de réductions adaptées à l'approximation différentielle ont été définies relativement tôt, la réduction affine en particulier, les résultats de complétude ont été obtenus assez récemment [6, 13].

Assez naturellement, l'étude de la structure de **NPO** pour l'approximation différentielle est « calquée » sur l'approximation standard, *i.e.*, complétude dans les classe **NPO**, **DAPX**, **DPTAS**, **Poly-DAPX**. En outre, la classe **0-DAPX**, propre au paradigme différentiel (voir [15] et section 3.2), est également intégrée à cette étude.

8.1 Complétude dans **NPO**

Faisant suite à **ORPONEN** et **MANNILA** ([53]), **AUSIELLO** *et al.* définissent la réduction **D**, analogue de la réduction stricte pour le rapport différentiel.

Définition 13. ([6]) Soient Π et Π' deux problèmes de **NPO**. On dit que Π se réduit à Π' sous la réduction **D** s'il existe deux fonctions f et g calculables en temps polynomial telles que :

- pour tout $I \in \mathcal{I}_\Pi$, $f(I) \in \mathcal{I}_{\Pi'}$;
- pour tout $I \in \mathcal{I}_\Pi$ et tout $x \in \text{sol}(f(I))$, $g(I, x) \in \text{sol}(\mathcal{I}_{\Pi'})$;
- pour tout $I \in \mathcal{I}_\Pi$ et tout $x \in \text{sol}(f(I))$, $\delta_\Pi(I, g(I, x)) \geq \delta_{\Pi'}(f(I), x)$. ■

Bien entendu, la réduction **D** préserve l'appartenance aux classes **DAPX** et **DPTAS**.

Par une démarche analogue à celles utilisées pour montrer leurs équivalents en approximation standard, *i.e.*, combinaison des preuves de complétude partielle (dans

Max-NPO et **Min-NPO**) et une équivalence entre les problèmes **MIN SAT PONDÉRÉ** et **MAX SAT PONDÉRÉ**, le résultat suivant est obtenu dans [6].

Théorème 26. ([6]) **MIN SAT PONDÉRÉ**, **MAX SAT PONDÉRÉ**, **MIN PL0-1** et **MAX PL0-1** sont **NPO-complets** sous la réduction D .

8.2 NPO-complétude et 0-DAPX

Comme nous l'avons déjà défini, une classe particulièrement naturelle pour l'approximation différentielle est la classe **0-DAPX**, celle des problèmes pour lesquels n'importe quel algorithme polynomial fournit une pire solution sur un ensemble infini d'instances. D'une certaine manière, les problèmes de cette classe apparaissent donc comme les plus difficiles à résoudre (selon le paradigme différentiel) de la classe **NPO**. Par ailleurs, la notion de complétude a pour vocation de refléter elle aussi la difficulté des problèmes. Vient alors une interrogation légitime. Les deux notions de difficulté ainsi définies sont-elles liées? Qu'en est-il de la relation entre problème de **0-DAPX** et problème **NPO-complet** sous la réduction D ? Le théorème suivant apporte une réponse à cette interrogation.

Théorème 27. ([6]) Sous la réduction D , **NPO-complet** \subseteq **0-DAPX**. En d'autres termes, tout problème **NPO-complet** est dans **0-DAPX**.

Ce résultat tend donc à confirmer l'idée selon laquelle le préordre entre problèmes induit par une réduction naturelle reflète une difficulté effective, opérationnelle et n'a donc pas seulement un sens structurel, théorique.

Il est à noter de plus que, si au lieu de D une réduction plus forte est utilisée, par exemple si on permet que au moins une des fonctions f et g dans la définition 13 soit multivaluée, alors, sous une telle réduction, la classe des problèmes **NPO-complets** coïncide avec **0-DAPX** ([6]).

8.3 Complétude dans DAPX

Il ressort des sections 5, 6 et 7 l'importance de la classe **APX** pour la problématique de la complétude : premiers résultats de complétude dans une classe d'approximation ([19]), étude dans une sous-classe définie syntaxiquement ([55]), premiers résultats de complétude en application du théorème **PCP** ([43]) et premiers problèmes intermédiaires naturels ([20])!

Il apparaît donc particulièrement intéressant d'étudier son équivalent différentiel, la classe **DAPX**. La première étape consiste à définir une réduction préservant l'appartenance à la classe **DPTAS**. Ausiello *et al.* ([6]) ont ainsi choisi de définir l'analogue dans le cadre différentiel de la réduction la plus large et certainement la plus étudiée, la réduction **PTAS**.

Définition 14. ([6]) Soient Π et Π' deux problèmes de **NPO**. On dit que Π se réduit à Π' sous la réduction **DPTAS** s'il existe trois fonctions f , g et c telles que :

- pour tout $I \in \mathcal{I}_\Pi$ et tout $\varepsilon \in]0,1[$, $f(I,\varepsilon) \in \mathcal{I}_{\Pi'}$; le temps de calcul de f est polynomial en fonction de $|I|$;

- pour tout $I \in \mathcal{I}_\Pi$, tout $\varepsilon \in]0,1[$ et tout $x \in \text{sol}(f(I))$, $g(I,\varepsilon,x) \in \text{sol}(\mathcal{I}_\Pi)$;
- le temps de calcul de g est polynomial en fonction de $|I|$ et de $|x|$;
- $c :]0,1[\rightarrow]0,1[$;
- pour tout $I \in \mathcal{I}_\Pi$, tout $x \in \text{sol}(f(I))$ et tout $\varepsilon \in]0,1[$: $\delta_{\Pi'}(f(I,\varepsilon),x) \geq 1 - c(\varepsilon)$ implique $\delta_\Pi(I,g(I,\varepsilon,x)) \geq 1 - \varepsilon$;
- la fonction f peut être multivaluée ; dans ce cas $f = (f_1, \dots, f_i)$, où i est polynomial en fonction de $|I|$, alors l'implication de l'item précédent devient : il existe $j \leq i$ tel que $(\delta_{\Pi'}(f_i(I,\varepsilon),x) \geq 1 - c(\varepsilon))$ implique $\delta_\Pi(I,g(I,\varepsilon,x)) \geq 1 - \varepsilon$. ■

Encore une fois, la réduction *DPTAS* préserve l'appartenance à la classe *DPTAS*.

Pour étudier la question de l'existence des problèmes complets dans **DAPX**, une première idée aurait été de reprendre la preuve de [19] pour obtenir la complétude de problèmes qui n'auraient malheureusement pas été naturels. Une approche potentiellement plus fructueuse consiste à vouloir réutiliser le théorème **PCP**. Or ce théorème est, jusqu'à présent, connecté uniquement à l'approximation standard, et il paraît alors peu évident d'obtenir grâce à lui des résultats de complétude en approximation différentielle. Il semble en définitive plus opportun de réutiliser les résultats de complétude dans **APX** pour en obtenir dans **DAPX**.

Le schéma de raisonnement est alors le suivant : pour prouver qu'un problème Π_1 est **DAPX**-complet, on cherche un problème Π_2 **APX**-complet qui se réduit à Π_1 sous une réduction *ad hoc* (transformant un schéma différentiel en un schéma standard). Ensuite, partant d'un problème Π_3 dans **DAPX**, on le réduit (c'est toujours possible) à un problème Π_4 de **APX** sous une autre réduction *ad hoc* (transformant cette fois un schéma standard en un schéma différentiel). On obtient finalement par transitivité une réduction de tout problème Π_3 de **DAPX** à Π_1 . En effet, le schéma différentiel (dont nous supposons l'existence) pour Π_1 fournit un schéma standard pour Π_2 , donc un schéma standard pour Π_4 , donc un schéma différentiel pour Π_3 : $\Pi_3 \leq \Pi_4 \leq \Pi_2 \leq \Pi_1$ (les réductions utilisées étant différentes). Par cette technique, le résultat suivant est obtenu dans [6].

Théorème 28. ([6]) *Les problèmes MAX STABLE-B, MIN COUVERTURE DE SOMMETS-B, MAX SET PACKING-B et MIN SET COVER-B sont DAPX-complets sous la réduction DPTAS.*

Le problème MAX STABLE-B est **APX**-complet ; de plus, pour ce problème, les rapports standard et différentiel coïncident. Tout le travail (que nous ne détaillerons pas ici) consiste donc à montrer que l'on peut réduire tout problème de **DAPX** à un problème de **APX** sous une réduction transformant un schéma standard en un schéma différentiel. Ceci permet d'obtenir la complétude de MAX STABLE-B. La complétude des autres problèmes s'en déduit alors par réduction depuis MAX STABLE-B.

Nous avons donc plusieurs problèmes naturels **APX**-complets, indiquant que la réduction *DPTAS* remplit le rôle pour lequel elle a été définie. Ces résultats, couplés à ceux de la complétude dans **NPO**, indiquent que la problématique de la complétude

apparaît aussi naturelle et aussi fructueuse dans le cadre de l'approximation différentielle que dans celui de l'approximation standard. Malgré l'importance du résultat du théorème 28, au moment où celui-ci a été produit nous pouvions ajouter deux bé-mols : (i) il n'est fait mention d'aucun résultat concernant l'existence de problèmes intermédiaires, naturels ou non ; (ii) les problèmes dont il est question dans le théorème 28 sont **APX**-complets ; il paraissait alors intéressant d'exhiber des problèmes **DAPX**-complets mais pas **APX**-complets. Même si le premier point reste toujours ouvert et d'actualité, le deuxième a été levé dans [13]. Il est démontré dans cette article que **MIN COLORATION** est **DAPX**-complet, alors qu'il n'est même pas dans **APX**.

8.4 Complétude dans **DPTAS**

Après les résultats de complétude dans **NPO** et **DAPX**, nous nous intéressons maintenant à la complétude dans la classe **DPTAS**. Ce problème a été abordé dans [6], par des réductions du type **KARP**, sans vraiment parvenir à une conclusion pleinement satisfaisante. Dans [13], la démarche décrite dans la section 5.3 a été étendue au paradigme différentiel. En effet, la réduction **FT** que nous y avons vue étant assez large, elle peut être réécrite en différentiel en remplaçant tout simplement la référence au rapport standard par une référence au rapport différentiel. Nous appelons **DFT** cette adaptation de la réduction **FT**. Il est immédiat de voir que la réduction **DFT** préserve l'appartenance à **DFPTAS**.

Par une démarche tout à fait analogue à celle de la section 5.3 le théorème suivant est démontré dans [13].

Théorème 29. ([13]) *Soit un problème $\Pi \in \mathbf{NPO}$ dont la version décision est **NP**-complète. Si Π est polynomialement borné au sens différentiel, alors tout problème $\Pi' \in \mathbf{NPO}$ se réduit à Π sous une réduction **DFT**. Par conséquent, si un problème de **NPO** est dans **DPTAS**, polynomialement borné au sens différentiel et **NP**-complet dans sa version décision, alors il est **DPTAS**-complet sous la réduction **DFT**.*

Il suffit maintenant d'observer que pour **MAX STABLE PLANAIRE** les rapports standard et différentiel coïncident et que **MAX STABLE PLANAIRE** \in **PTAS** ([12]) ; par ailleurs, **MIN COUVERTURE DE SOMMETS PLANAIRE** est aussi dans **DPTAS**, par réduction affine au problème **MAX STABLE PLANAIRE** ; enfin, le problème **BIN PACKING** est dans **DPTAS** ([24]). Ces trois problèmes étant polynomialement bornés au sens différentiel, nous en déduisons qu'ils sont **DPTAS**-complets.

Théorème 30. ([13]) *Les problèmes **MAX STABLE PLANAIRE**, **MIN COUVERTURE DE SOMMETS PLANAIRE** et **BIN PACKING** sont **DPTAS**-complets sous la réduction **DFT**.*

Notons enfin que le même résultat sur les conditions d'existence des problèmes intermédiaires (la contrepartie différentielle du théorème 12) reste en vigueur pour la réduction **DFT**.

8.5 Complétude dans Poly-DAPX

Nous complétons cet état de l'art sur le carrefour de la complexité avec l'approximation polynomiale en abordant la complétude dans la classe **Poly-DAPX**. Dans [13] où cette question a été étudiée, la réduction DPTAS a été choisie pour que la question qui nous occupe ici soit appréhendée. Notons que nous devons de toute façon utiliser une réduction qui ne préserve pas l'optimalité des solutions, pour ne pas être confronté au même problème évoqué précédemment dans la section 7.3.2.

Remarquons aussi que l'on peut se limiter aux problèmes de maximisation. En effet, si nous prenons un problème de minimisation Π , définissons le problème Π' comme étant un problème de maximisation avec les mêmes instances et où la fonction objectif est $m_{\Pi'}(I,x) = M - m_{\Pi}(I,x)$ (où M est un majorant de la valeur des solutions de l'instance I). Les deux problèmes sont affinements équivalents (donc en particulier Π se DPTAS-réduit à Π'). De plus, si $\Pi \in \mathbf{Poly-DAPX}$, alors $\Pi' \in \mathbf{Poly-DAPX}$. Dans la suite, nous considérerons donc uniquement le cas de problèmes de maximisation. Nous pouvons alors énoncer le théorème suivant.

Théorème 31. ([13]) *Si Π' est un problème canoniquement difficile pour la classe **Poly-DAPX**, alors tout problème de **Poly-DAPX** se DPTAS-réduit à Π' .*

Le lecteur a noté que dans l'énoncé du théorème 31, nous nous sommes passés de l'hypothèse d'additivité, nécessaire dans le cas de l'approximation standard sous la réduction PTAS. Cela fut possible grâce au fait que la réduction DPTAS peut être multivaluée. En outre, si nous permettions à une réduction PTAS d'être multivaluée, nous pourrions tout à fait nous passer d'additivité également dans le théorème 21 de la section 7.3.2.

Etant donné que pour MAX STABLE et MAX CLIQUE les rapports standard et différentiel coïncident et que ces deux problèmes sont dans **Poly-APX** (section 7.3.2), et que MIN COUVERTURE DE SOMMETS est équivalent, par réduction affine à MAX STABLE, le résultat suivant est immédiatement vrai.

Théorème 32. MAX STABLE, MAX CLIQUE et MIN COUVERTURE DE SOMMETS sont **Poly-DAPX-complets** sous la réduction DPTAS.

9 Discussion

Le programme de recherche qui vise à transposer les notions de réduction et de complétude des problèmes de décision aux problèmes d'optimisation, est un vaste programme, très actif depuis trente ans. Plusieurs notions de réductions ont été définies et une multitude des résultats les utilisant ont vu le jour jusqu'aujourd'hui. Cet état de l'art n'a pas présenté toutes ces réductions malgré l'intérêt évident de chacune d'elle. Il s'est contenté de celles par le biais desquelles une réelle structuration du **NPO** (par des résultats de complétude) est rendue possible.

Ces réductions génériques, structurant les classes d'approximation, sont d'abord basées sur des adaptations de la preuve du théorème de COOK. Ces recherches ont

abouti à des résultats très satisfaisants, tout en ne parvenant pas à résoudre des questions fondamentales, en particulier la complétude dans **APX** d'un problème comme **MAX 3SAT**. L'outil **PCP** a alors permis de renforcer le lien entre inapproximabilité et complétude en donnant une méthode générique permettant d'aboutir à des résultats de complétude à partir de résultats négatifs. **TREVISAN** ([62]) a même commenté cette avancée spectaculaire en disant que **KHANNA et al.** « ont donné une réponse définitive à la question de la complétude dans les classes d'approximation ».

Les résultats présentés dans cet état de l'art concernent les classes où le degré d'approximation est constant ou fonction de l'instance. Nous pouvons cependant envisager des notions de complétude dans d'autres classes d'approximation. **AUSIELLO et al.** définissent par exemple dans [11] une réduction préservant non seulement l'approximation mais également la qualité des optima locaux. Ils obtiennent alors pour une classe de problèmes admettant de « bons » optima locaux des problèmes complets sous la réduction qu'ils ont définie. De telles études structurelles, dont un début est présenté dans [60], sont à reprendre dans le cadre de la théorie de l'approximation différentielle. Un autre travail consiste à raffiner l'étude de la réduction **E** par déterminer s'il existe ou non des problèmes intermédiaires. La réduction **F** introduite dans [19] a certes été la première à introduire une notion de **PTAS**-complétude, mais seul un problème artificiel a été montré complet. Peut-on trouver des problèmes **PTAS**-complets naturels sous cette réduction ?

Par ailleurs, et c'est peut-être là le sens du jugement de **TREVISAN**, on peut s'apercevoir que les réductions génériques basées sur le théorème de **COOK** ou sur le théorème **PCP** n'ont apporté aucune information sur l'inapproximabilité des problèmes. Elles présentent un intérêt structurel, permettent de lier des problèmes d'optimisation entre eux, mais l'inapproximabilité générée par les résultats de complétude était déjà connue. Cet état de fait paraît intrinsèque aux réductions basées sur le théorème **PCP** (puisque l'on part d'un résultat négatif), plus décevant pour les réductions basées sur la preuve du théorème de Cook. Il semble donc que les réductions spécifiques entre problèmes particuliers permettent de transférer des résultats (souvent négatifs) mais que l'intérêt de la complétude vis-à-vis de l'inapproximabilité est moins évident.

Dans cet ordre d'idée, les résultats concernant **Max-SNP** apparaissent singuliers. En effet, la réduction proposée par **PAPADIMITRIOU** et **YANNAKAKIS** a deux spécificités intéressantes. Premièrement, elle est en un certain sens plus « constructive » que les autres (une construction effective liée au théorème **PCP** semble par exemple vraiment utopique). Deuxièmement, comme nous l'avons déjà mentionné, la **Max-SNP**-complétude de **MAX 3SAT** s'inscrit vraiment dans l'esprit de la **NP**-complétude. Si ce résultat ne répond pas directement à la question de l'existence de schéma pour ce problème, il indique néanmoins qu'en trouver un permettrait d'en trouver un pour tous les problèmes de **Max-SNP**. Cela représentait donc une avancée pour l'inapproximabilité de **MAX 3SAT**.

Partant de ce constat, nous pouvons alors nous demander si la définition structurée des problèmes sous forme logique telle qu'elle se présente, par exemple dans [31, 44, 45], mais qui n'est pas présentée dans cet état de l'art, ne pourrait pas s'avérer utile dans la thématique de la complétude.

Références

- [1] A. Aiello, E. Burattini, M. Furnari, A. Massarotti, and F. Ventriglia. Computational complexity: the problem of approximation. In J. Demetrovic, G. Katona, and A. Saloana, editors, *Algebra, combinatorics, and logic in computer science*, volume I, pages 51–62, New York, 1986. Colloquia Mathematica Societatis János Bolyai, North-Holland.
- [2] S. Arora and C. Lund. Hardness of approximation. In D. S. Hochbaum, editor, *Approximation algorithms for NP-hard problems*, chapter 10, pages 399–446. PWS, Boston, 1997.
- [3] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. In *Proc. FOCS'92*, pages 14–23, 1992.
- [4] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. *J. Assoc. Comput. Mach.*, 45(3):501–555, 1998.
- [5] S. Arora and M. Sudan. Improved low-degree testing and its applications. *Combinatorica*, 23(3):365–426, 2003.
- [6] G. Ausiello, C. Bazgan, M. Demange, and V. Th. Paschos. Completeness in differential approximation classes. *International Journal of Foundations of Computer Science*, 16(6):1267–1295, 2005.
- [7] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and approximation. Combinatorial optimization problems and their approximability properties*. Springer-Verlag, Berlin, 1999.
- [8] G. Ausiello, P. Crescenzi, and M. Protasi. Approximate solutions of NP optimization problems. *Theoret. Comput. Sci.*, 150:1–55, 1995.
- [9] G. Ausiello, A. D’Atri, and M. Protasi. On the structure of combinatorial problems and structure preserving reductions. In *Proc. ICALP’77*, volume 52 of *Lecture Notes in Computer Science*, pages 45–60. Springer-Verlag, 1977.
- [10] G. Ausiello, A. D’Atri, and M. Protasi. Structure preserving reductions among convex optimization problems. *J. Comput. System Sci.*, 21:136–153, 1980.
- [11] G. Ausiello and M. Protasi. NP optimization problems and local optima graph theory. In Y. Alavi and A. Schwenk, editors, *Combinatorics and applications. Proc. Quadriennial International Conference on the Theory and Applications of Graphs*, volume 2, pages 957–975. Wiley and Sons, 1995.
- [12] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. Assoc. Comput. Mach.*, 41(1):153–180, 1994.
- [13] C. Bazgan, B. Escoffier, and V. Th. Paschos. Completeness in standard and differential approximation classes: Poly-(D)APX- and (D)PTAS-completeness. *Theoret. Comput. Sci.*, 339:272–292, 2005.
- [14] C. Bazgan, R. Hassin, and J. Monnot. Approximation algorithms for some vehicle routing problems. *Discrete Appl. Math.*, 146(1):27–42, 2005.

- [15] C. Bazgan and V. Th Paschos. Differential approximation for optimal satisfiability and related problems. *European J. Oper. Res.*, 147(2):397–404, 2003.
- [16] P. Berman and G. Schnitger. On the complexity of approximating the independent set problem. *Information and Computation*, 96:77–94, 1992.
- [17] S. A. Cook. The complexity of theorem-proving procedures. In *Proc. STOC'71*, pages 151–158, 1971.
- [18] P. Crescenzi, V. Kann, R. Silvestri, and L. Trevisan. Structure in approximation classes. *SIAM J. Comput.*, 28(5):1759–1782, 1999.
- [19] P. Crescenzi and A. Panconesi. Completeness in approximation classes. *Information and Computation*, 93(2):241–262, 1991.
- [20] P. Crescenzi, R. Silvestri, and L. Trevisan. Natural complete and intermediate problem in approximation classes. Manuscript, 1994.
- [21] P. Crescenzi and L. Trevisan. Max NP-completeness made easy. *Theoret. Comput. Sci.*, 225(1-2):65–79, 1999.
- [22] P. Crescenzi and L. Trevisan. On approximation scheme preserving reducibility and its applications. *Theory of Computing Systems*, 33(1):1–16, 2000.
- [23] M. Demange, P. Grisoni, and V. Th. Paschos. Differential approximation algorithms for some combinatorial optimization problems. *Theoret. Comput. Sci.*, 209:107–122, 1998.
- [24] M. Demange, J. Monnot, and V. Th. Paschos. Bridging gap between standard and differential polynomial approximation: the case of bin-packing. *Appl. Math. Lett.*, 12:127–133, 1999.
- [25] M. Demange, J. Monnot, and V. Th. Paschos. Maximizing the number of unused bins. *Found. Comput. Decision Sci.*, 26(2):169–186, 2001.
- [26] M. Demange and V. Th. Paschos. Quelques étapes vers la conciliation de la théorie d'approximation et celle d'optimisation : une nouvelle théorie d'approximation polynomiale et résultats préliminaires. *C. R. Acad. Sci. Paris Sér. I Math.*, 317:409–414, 1993.
- [27] M. Demange and V. Th. Paschos. On an approximation measure founded on the links between optimization and polynomial approximation theory. *Theoret. Comput. Sci.*, 158:117–141, 1996.
- [28] I. Dinur. The PCP theorem by gap amplification. In *Proc. STOC'06*, pages 241–250, 2006.
- [29] I. Dinur. The PCP theorem by gap amplification. Unpublished manuscript, 2007.
- [30] R. Duh and M. Fürer. Approximation of k -set cover by semi-local optimization. In *Proc. STOC'97*, pages 256–265, 1997.
- [31] B. Escoffier and V. Th. Paschos. Proving completeness by logic. *International Journal of Computer Mathematics*, 82(2):151–161, 2005.
- [32] B. Escoffier and V. Th. Paschos. Completeness in approximation classes beyond apx. *Theoret. Comput. Sci.*, 359(1–3):369–377, 2006.

- [33] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In R. M. Karp, editor, *Complexity of computations*, pages 43–73. American Mathematics Society, 1974.
- [34] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. Assoc. Comput. Mach.*, 45:634–652, 1998.
- [35] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive proofs and the hardness of approximating cliques. *J. Assoc. Comput. Mach.*, 43(2):268–292, 1996.
- [36] M. R. Garey and D. S. Johnson. *Computers and intractability. A guide to the theory of NP-completeness*. W. H. Freeman, San Francisco, 1979.
- [37] R. Hassin and S. Khuller. z -approximations. *J. Algorithms*, 41:429–442, 2001.
- [38] R. Hassin and J. Monnot. The maximum saving partition problem. *Oper. Res. Lett.*, 33:242–248, 2005.
- [39] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.
- [40] J. Håstad. Some optimal inapproximability results. *J. Assoc. Comput. Mach.*, 48:798–859, 2001.
- [41] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. System Sci.*, 9:256–278, 1974.
- [42] V. Kann. Polynomially bounded problems that are hard to approximate. *Nordic J. Computing*, 1:317–331, 1994.
- [43] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani. On syntactic versus computational views of approximability. *SIAM J. Comput.*, 28:164–191, 1998.
- [44] Ph. G. Kolaitis and M. N. Thakur. Logical definability of NP-optimization problems. *Information and Computation*, 115(2):321–353, 1994.
- [45] Ph. G. Kolaitis and M. N. Thakur. Approximation properties of NP minimization classes. *J. Comput. System Sci.*, 50(3):391–411, 1995.
- [46] M. W. Krentel. The complexity of optimization problems. *J. Comput. System Sci.*, 36:490–509, 1988.
- [47] R. E. Ladner. On the structure of polynomial time reducibility. *J. Assoc. Comput. Mach.*, 22:155–171, 1975.
- [48] R. E. Ladner, N. A. Lynch, and A. L. Selman. A comparison of polynomial time reducibilities. *Theoret. Comput. Sci.*, 1:103–123, 1975.
- [49] H. R. Lewis and C. H. Papadimitriou. *Elements of the theory of computation*. Prentice-Hall, 1981.
- [50] J. Monnot, V. Th. Paschos, and S. Toulouse. Differential approximation results for the traveling salesman problem with distances 1 and 2. *European J. Oper. Res.*, 145(3):557–568, 2002.
- [51] J. Monnot, V. Th. Paschos, and S. Toulouse. Approximation algorithms for the traveling salesman problem. *Mathematical Methods of Operations Research*, 57(1):387–405, 2003.

- [52] J. Monnot, V. Th. Paschos, and S. Toulouse. Optima locaux garantis pour l'approximation différentielle. *Technique et Science Informatiques*, 22(3):257–288, 2003.
- [53] P. Orponen and H. Mannila. On approximation preserving reductions: complete problems and robust measures. Technical Report C-1987-28, Dept. of Computer Science, University of Helsinki, Finland, 1987.
- [54] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.
- [55] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *J. Comput. System Sci.*, 43:425–440, 1991.
- [56] V. Th. Paschos. *Complexité et approximation polynomiale*. Hermès, Paris, 2004.
- [57] A. Paz and S. Moran. Non deterministic polynomial optimization problems and their approximations. *Theoret. Comput. Sci.*, 15:251–277, 1981.
- [58] R. Raz and S. Safra. A sub-constant error probability low-degree test and a sub-constant error probability PCP characterization of NP. In *Proc. STOC'97*, pages 475–484, 1997.
- [59] P. Slavík. A tight analysis of the greedy algorithm for set cover. In *Proc. STOC'96*, pages 435–441, 1996.
- [60] S. Toulouse. *Approximation polynomiale: optima locaux et rapport différentiel*. Thèse de doctorat, LAMSADE, Université Paris-Dauphine, 2001.
- [61] L. Trevisan. Inapproximability of combinatorial optimization problems. The Computing Research Repository, 2004.
- [62] L. Trevisan. Inapproximabilité des problèmes d'optimisation combinatoire. In V. Th. Paschos, editor, *Optimisation combinatoire : concepts avancés*, chapter 3, pages 91–138. Hermès Science, Paris, 2005. Traduction française de [61] par B. Escoffier.
- [63] V. Vazirani. *Approximation algorithms*. Springer, Berlin, 2001.
- [64] E. Zemel. Measuring the quality of approximate solutions to zero-one programming problems. *Math. Oper. Res.*, 6:319–332, 1981.