MATHÉMATIQUES ET SCIENCES HUMAINES

MARC DEMANGE
VANGELIS TH. PASCHOS

Valeurs extrémales d'un problème d'optimisation combinatoire et approximation polynomiale

Mathématiques et sciences humaines, tome 135 (1996), p. 51-66
<http://www.numdam.org/item?id=MSH_1996__135__51_0>

Math. Inf. Sci. hum. (34^{e} année, n°135, *1996*, pp.51-66)

VALEURS EXTRÉMALES D'UN PROBLÈME D'OPTIMISATION COMBINATOIRE ET APPROXIMATION POLYNOMIALE

Marc DEMANGE[1, 2] ; Vangelis Th. PASCHOS[2, 3]

RÉSUMÉ — *A la suite de quelques-uns de nos travaux antérieurs sur la théorie de la complexité et de l'approximation polynomiale, nous présentons quelques nouvelles réflexions et arguments sur les valeurs (et solutions) extrémales, (optimale et pire), des problèmes d'optimisation combinatoire. Cette discussion nous conduit à considérer la limite entre constructibilité et non-constructibilité, source constante de contradiction en théorie de la complexité. En effet, cette théorie, telle qu'on la connaît et manie aujourd'hui, est fondée sur la non-constructibilité tandis que deux de ses domaines principaux, l'optimisation combinatoire et la théorie de l'approximation polynomiale, nécessitent un cadre conceptuel fondé sur la constructibilité. L'approximation polynomiale dépasse aujourd'hui sa conception originelle (en tant qu'ensemble d'outils pour la résolution rapide des problèmes* NP-*complets), intervient très fortement dans la définition de nouvelles notions et objets mathématiques et fait ainsi partie à part entière de "l'arsenal" de la complexité. Elle est un outil théorique majeur pour l'appréhension, l'approfondissement et l'enrichissement de la théorie de la complexité et notamment de la connaissance de la classe* NP. *Ces développements récents de l'approximation polynomiale dévoilent des problèmes particulièrement intéressants, notamment d'un point de vue épistémologique.*

SUMMARY — Extremal values of a combinatorial optimization problem and polynomial approximation
As a subsequence of some of our previous works on complexity and polynomial approximation theory, we present some further reflections and arguments about extremal, optimal and worst, values (and solutions) of combinatorial optimization problems. This discussion leads us to consider a constant source of contradictions in complexity theory, the limits between constructibility and non-constructibility. In fact, complexity theory, in its current form, is founded on non-constructibility while, two of the main of its topics, the combinatorial optimization and the polynomial approximation theory need both a conceptual framework founded on constructibility. Approximation theory today goes beyond its framework of origin (a set of tools for finding fast solutions for NP-*complete problems) since it strongly intervenes in the definition of new mathematical notions and objects making entirely part of the "arsenal" of complexity and it constitutes a major theoretical tool as well for understanding, deepening and enriching complexity theory as for better apprehending class* NP. *This recent "problemshift" for the polynomial approximation theory brings to the fore new and particularly interesting problems from both mathematical and epistemological points of view.*

[1] LAMSADE, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16, France {demange, paschos}@lamsade.dauphine.fr.
[2] CERMSEM, Université Paris I, 90 rue de Tolbiac, 75634 Paris Cedex 13, demange@univ-paris1.fr.
[3] Auteur pour toute correspondance concernant l'article.

1 INTRODUCTION

Rappelons que la théorie de la complexité a pour principal objet l'étude des problèmes NP qui sont classiquement définis ([13]) comme des problèmes de décision (chaque instance est une question) résolus par une machine de Turing non déterministe polynomiale[1]. On distingue essentiellement deux familles de problèmes au sein de NP, la classe P des problèmes *polynomiaux* et la classe NP-C des problèmes *NP-complets*. Pour fixer les idées, ces deux types de problèmes peuvent être respectivement considérés comme les problèmes les plus faciles et les plus difficiles de NP. Par définition, tout problème P peut être résolu en temps polynomial par une machine de Turing déterministe. Par contre, les problèmes NP-complets sont définis de telle sorte que l'existence d'une machine de Turing déterministe polynomiale résolvant l'un d'entre eux impliquerait l'existence d'une telle machine pour n'importe quel problème NP. Ainsi, la fameuse condition P$\neq$NP est bien entendu équivalente à P$\neq$NP-C ou encore P$\cap$NP-C$= \emptyset$. Cette hypothèse, qui signifie intuitivement qu'un problème NP-Complet n'est pas résoluble en temps polynomial, est à la base de tous les travaux de complexité si bien que la réflexion qui vous est contée içi pourrait commencer par: *si P était différent de NP, alors ...*

Cette hypothèse étant faite, l'une des démarches classiques ([13,4]), appelée *approximation polynomiale*, consiste à étudier dans quelle mesure et en quel sens des problèmes NP-Complets peuvent être résolus de façon approchée, en temps polynomial et avec des garanties absolues sur la qualité des solutions obtenues. Cette exigence, au niveau des garanties de qualité et de complexité des algorithmes étudiés, différencie ce point de vue d'une étude classique par heuristiques.

En fait, même si son but originel est d'offrir une certaine possibilité de résolution efficace de problèmes NP-complets, l'approximation polynomiale nécessite un cadre formel légèrement différent de celui de la théorie de la complexité. En effet, cette dernière a été conçue ([13]) dans le cadre de problèmes de décision alors que l'approximation a besoin d'un cadre d'optimisation, tant pour donner un sens à la notion de solution approchée que pour mesurer la qualité de telles solutions. La définition suivante (cf. [9]) délimite le cadre d'étude de l'approximation et définit ses liens avec la théorie de la complexité.

DEFINITION 1. Soit opt $\in \{\max, \min\}$. Un "problème d'optimisation combinatoire affine élémentaire" (ou problème élémentaire d'optimisation affine (0,1)) est un quadruplet (n, t, v, C), où $n \in I\!N$ est la taille du problème élémentaire, t est une dimension, $v(\cdot)$ est une fonction affine sur $I\!R^t$ et C est un polytope de $I\!R^t$.
Nous appelons un tel problème élémentaire "instance de taille n" (notée I) et nous l'exprimons sous la forme

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

Un "problème d'optimisation combinatoire NP (resp. NP-complet)" est un ensemble d'instances tel que l'ensemble correspondant des questions "pour un M donné, existe-t-il $\vec{x} \in C$ tel que $v(\vec{x})\theta M$?" (où θ vaut $\geq$ si opt vaut max et $\leq$ si opt vaut min) est exactement l'ensemble des instances d'un problème NP (resp. NP-complet) au sens usuel de la théorie des langages ([13]). ∎

En fait, remarquons que seuls certains problèmes NP-Complets admettent une version optimisation; ce sont ces versions optimisation que traite l'approximation polynomiale. Pour un tel problème, un algorithme approché est, par définition, un algorithme polynomial fournissant, pour chaque instance I, une solution réalisable (i.e. de C) de valeur objective $\lambda(I)$.

[1] Plus formellement, la *résolution* d'un tel problème signifie qu'un langage qui représente les instances positives du problème de décision est accepté par la machine de Turing.

La grande majorité des travaux d'approximation polynomiale utilisent, pour mesurer la qualité de la solution obtenue pour I, la quantité $\lambda(I)/\beta(I)$ où $\beta(I)$ désigne la valeur optimale de l'instance. Un résultat d'approximation consisté alors à borner ce rapport pour toute instance du problème étudié. Nous désignons par ***classique*** l'ensemble de la théorie fondée sur ce rapport d'approximation.

Outre ses applications pratiques quant à la résolution effective de problèmes difficiles, l'approximation polynomiale a d'importants enjeux théoriques au niveau de la compréhension de la classe NP-C et soulève de nombreux problèmes mathématiques et épistémologiques. Dans cet article, nous nous proposons d'étudier certaines de ces questions qui apparaissent dans la continuité d'un travail récent ([8,9] - voir également [7]), dans lequel nous avons développé une nouvelle théorie d'approximation polynomiale fondée sur la notion de rapport d'approximation différentiel.

Sans entrer dans les détails de ce travail, nous avons besoin d'en esquisser les principales idées afin de rendre la discussion qui suit claire et pertinente. Il s'agissait de répondre à plusieurs paradoxes de la théorie classique illustrant une certaine incompatibilité de cette dernière avec la théorie de l'optimisation combinatoire qui fournit pourtant, comme nous venons de le rappeler, le cadre d'étude adéquat à l'approximation polynomiale. En effet, la théorie classique de l'approximation polynomiale a adopté, par sa nature, le cadre de l'optimisation sans intégrer toute la logique qu'il implique ni analysé précisément la portée d'un tel choix.

Une réflexion sur le concept de rapport d'approximation, argumentée d'une part par l'analyse d'aspects paradoxaux de la théorie classique et d'autre part par une démarche axiomatique, nous a conduits à la notion de rapport d'approximation différentiel qui consiste à mesurer la position de la valeur atteinte $\lambda(I)$ dans l'intervalle des valeurs possibles. Pour une instance I, ce rapport s'exprime par:

$$\rho_{\mathcal{A}}(I) = \frac{\omega(I) - \lambda(I)}{\omega(I) - \beta(I)}$$

où $\omega(I)$ désigne la valeur d'une pire solution définie de la façon suivante.

DEFINITION 2. Soit un problème d'optimisation combinatoire Π au sens de la définition 1; pour toute instance I s'exprimant

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

on appelle pire solution de I toute solution optimale du programme suivant, instance de $\bar{\Pi}$:

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

où $\overline{\text{opt}}$ = min(max) si opt = max(min). La pire valeur $\omega(I)$ est la valeur objective d'une pire solution. ∎

Dans l'article [9], l'approche axiomatique est essentiellement fondée sur une notion d'équivalence entre problèmes d'optimisation combinatoire NP-complets; nous démontrons qu'une mesure fonction seulement des valeurs optimale et approchée ne peut vérifier l'axiome principal, d'où la nécessité de mesurer la qualité d'un algorithme approché par rapport à un troisième paramètre. Plusieurs raisons nous on alors menés à proposer la pire valeur comme paramètre supplémentaire. Puis, à partir de cette forme générale d'un rapport d'approximation, fonction des valeurs approchée, optimale et pire, nous développons la démarche axiomatique conduisant au rapport différentiel. Les deux valeurs réalisables extrèmes d'un problème combinatoire, la valeur optimale et la pire valeur, jouent donc un rôle essentiel dans le cadre de la théorie du rapport différentiel puisqu'elles servent de base à la mesure de la qualité d'un algorithme approché, et ce, non seulement dans la construction et justification du rapport différentiel, mais aussi surtout dans les nombreux résultats d'approximation établis dans ce cadre ([8,7]) ainsi que dans les algorithmes associés à ces résultats.

Mesurer la qualité d'un algorithme approché par rapport à la valeur optimale semble incontournable et est en tout cas universellement admis (nous verrons d'ailleurs que cette remarque naïve cache en fait tout un panel de problèmes au cœur des recherches actuelles), par contre la mesurer aussi par rapport à la pire valeur est propre à la notion de rapport différentiel. Ce choix a été déjà largement justifié dans les différents travaux sur la définition de ce rapport d'approximation ([9]) en réponse aux différents problèmes que nous avions soulevés dans la théorie classique; notre but n'est pas de reprendre ici ces arguments.

Indépendamment de la problématique à l'origine de la théorie du rapport d'approximation différentiel, nous présentons dans cet article quelques points de réflexion concernant les valeurs extrémales (optimale et pire) d'un problèmes d'optimisation combinatoire, et notamment leurs liens avec l'approximabilité du problème. Ce travail contribue à mettre en évidence la très grande richesse de ces deux notions et leur pertinence dans le cadre de l'approximation polynomiale des problèmes NP-complets. Il soulève également plusieurs questions que nous discutons et dont l'intérêt et les enjeux dépassent, à notre avis, le cadre spécifique de l'approximation polynomiale mais peuvent également intéresser, déjà d'un point de vue épistémologique, des mathématiciens non spécialistes de ce domaine.

Tout d'abord, le choix d'envisager la notion de pire valeur pour évaluer la qualité d'une solution approchée est relativement contre-intuitif. Pourtant l'ensemble des arguments présentés dans cet article et dans nos travaux antérieurs tendent à justifier cet usage. Par ailleurs, cette discussion nous mènera à une autre problématique qui intéresse depuis peu les théoriciens de l'approximation polynomiale. Il s'agit des liens entre les points de vue constructif et non-constructif. Cette question revient à étudier la difficulté relative entre la détermination, exacte ou approchée, de la valeur optimale d'un problème et d'une (ou de l'ensemble des) solution(s) réalisant cette valeur. Cette problématique va même jusqu'à remettre en cause les notions de problème et d'algorithme. Enfin, une question ouverte qui se pose naturellement dans la continuité des travaux évoqués précédemment serait: *peut-on, au moins partiellement, inclure dans l'axiomatique le choix de ces paramètres pour l'évaluation d'un rapport d'approximation?*

2 LA PIRE VALEUR D'UN PROBLEME

Une première remarque concernant la pire valeur d'un problème porte sur la complexité de son évaluation. En effet, les auteurs de l'article [3] l'appellent *valeur triviale*; il ne faudrait cependant pas en conclure qu'elle est toujours simple à calculer. Si c'est effectivement le cas pour certains problèmes, pour d'autres, au contraire, le calcul de la pire valeur est NP-complet. Considérons par exemple le problème de recherche, dans un graphe, d'un stable maximal pour l'inclusion de taille minimum, connu comme étant NP-complet ([13]); l'évaluation de sa pire valeur l'est aussi, puisqu'elle correspond au calcul du nombre de stabilité du graphe. Un autre exemple bien connu concerne le problème — polynomial — de recherche d'un couplage maximal de taille maximum (c'est à dire simplement la recherche d'un couplage maximum). Sa pire valeur est la cardinalité d'un couplage maximal pour l'inclusion de taille minimum qui est NP-complet à déterminer ([13]). En fait, les exemples foisonnent, ce qui est très naturel vu la définition de la pire valeur d'un problème Π comme valeur optimale du problème $\bar{\Pi}$.

2.1 Pire valeur et formulation du problème

Les deux exemples précédents illustrent un point tout à fait essentiel; pourquoi ne pas considérer que le problème de recherche d'un stable maximal de cardinalité maximum n'est pas simplement le problème de recherche d'un stable maximum, problème dont la pire valeur est nulle? Nous tenterons de répondre à cette question dans le prochain paragraphe, contentons-nous pour l'instant de noter qu'il serait aberrant que la notion de pire puisse être aussi extensible et qu'en fonction de l'interprétation du problème sa pire valeur et même la complexité de calcul de cette dernière soit à ce point modifiées. Force nous est donc d'admettre que les deux problèmes — recherche d'un stable maximum — et — recherche d'un stable maximal maximum — sont différents. Pour s'en convaincre, examinons ces deux problèmes du point de vue de la définition 2. Une instance générique du problème de stable maximum est:

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

où A est la matrice arêtes-sommets du graphe pris comme instance, la pire valeur est par définition la valeur optimale du problème

$$\bar{\mathrm{S}} = \begin{cases} \min & \vec{1}_{|V|} \cdot \vec{x} \\ & A \cdot \vec{x} \leq \vec{1}_{|E|} \\ & \vec{x} \in \{0,1\}^{|V|} \end{cases}$$

qui est donc nulle, ce qui correspond à la solution $x = \vec{0}$, i.e., à une solution vide. Quant au second problème, il se formule comme suit:

$$\mathrm{S}' = \begin{cases} \max & \vec{1}_{|V|} \cdot \vec{x} \\ & A \cdot \vec{x} \leq \vec{1}_{|E|} \\ & B \cdot \vec{x} \geq \vec{1}_{|V|} \\ & \vec{x} \in \{0,1\}^{|V|} \end{cases}$$

où A est la matrice arêtes-sommets du graphe et B sa matrice sommets-sommets. La contrainte rajoutée exprime que $\vec{x}$, qui est stable d'après la première contrainte, doit être maximal pour l'inclusion. La pire valeur de ce problème est bien la valeur d'un stable maximal minimum.

Ainsi, la spécification du pire est étroitement liée à la description complète et claire des contraintes du problème envisagé sous l'angle de la définition 1. Réfléchir à cette pire valeur contribue à formuler correctement le problème de sorte qu'apparaissent intrinsèquement à cette formulation les conditions qu'on désire imposer aux solutions réalisables. C'est l'objet même de la définition 1 que d'inclure l'ensemble du problème dans sa formulation; nous verrons dans le prochain paragraphe combien ceci est essentiel en approximation alors que le cadre de la résolution exacte "tolère" un certain flou un peu trop tentant.

Pour plusieurs problèmes même, la notion de pire doit être envisagée en amont de la formulation en termes de programme linéaire car elle lui est nécessaire. Considérons par exemple le problème de coloration; il s'agit, étant donné un graphe $G = (V, E)$, de partitionner ses sommets en un minimum d'ensembles stables correspondant chacun à une couleur. Il est très naturel de considérer comme objectif le nombre de couleurs utilisées, donc de définir comme variables intervenant dans l'objectif, un vecteur $\vec{y}$ à coordonnés (0,1) dont chaque "1" correspondrait à une couleur choisie (l'objectif serait alors $\vec{1} \cdot \vec{y}$). Cependant, pour qu'une telle formulation soit possible, il faut bien déterminer un espace ambiant pour $\vec{y}$, c'est à dire une dimension. Pour cela, il faut une connaissance a priori sur le problème de coloration: savoir qu'on n'aura jamais besoin de plus de l couleurs. En effet, avoir une borne calculable en temps polynomial pour la pire valeur, permet de définir la dimension l de $\vec{y}$ et de formuler le problème de la façon suivante:

$$\left\{\begin{array}{lll} \min & \vec{1}_l \cdot \vec{y} & \\ & A\vec{x}_i \leq \vec{1}_{|E|} & i \in \{1,\ldots,l\} \\ & \vec{x}_i - y_i\vec{1}_{|V|} \leq \vec{0}_{|V|} & i \in \{1,\ldots,l\} \\ & \sum_{i=1}^{l} \vec{x}_i = \vec{1}_{|V|} & \\ & \vec{y} \in \{0,1\}^l & \\ & \vec{x}_i \in \{0,1\}^{|V|} & i \in \{1,\ldots,l\} \end{array}\right.$$

où A est la matrice arêtes-sommets de G, les l variables $\vec{x}_i$ sont toutes de dimension $|V|$ et correspondent aux ensembles, éventuellement vides, de sommets de même couleur. On distingue quatre blocs de contraintes: les l contraintes de stabilité des $\vec{x}_i$, les l contraintes d'exclusion signifiant que, si la couleur i n'est pas choisie, le stable $\vec{x}_i$ est vide, et enfin les contraintes (0,1) classiques pour les vecteurs caractéristiques.

On peut choisir par exemple $l = |V|$, ce qui revient à considérer qu'il n'y a de toute façon pas plus de couleurs que de sommets dans G et qui constitue bien une connaissance a priori, si faible soit-elle, sur la valeur du problème. Remarquons que dès que la borne l dépasse le nombre chromatique du graphe, elle est réalisable et correspond exactement à la pire valeur du problème induit. Dans le cas où on choisit $l = |V|$, cette pire valeur est calculable en temps linéaire.

Les problèmes, tels que nous les définissons, correspondent souvent à sélectionner certains éléments dans un ensemble (on obtient alors un problème en variables (0,1)). Ainsi, chercher un stable, c'est chercher une partie de V, chercher un couplage, c'est chercher une partie de E. Pour les problèmes de minimisation, la taille de cet ensemble donne une borne pour la pire valeur. Mais un tel ensemble où sélectionner n'est pas toujours aussi naturel; il faut alors, comme pour le problème de coloration, utiliser une certaine connaissance du problème pour le définir.

Le problème de couverture minimum d'ensemble montre bien cet aller-retour entre la formulation et la notion de pire. Il s'agit, étant donné une famille d'ensembles $\mathcal{S} = \{S_1,\ldots,S_n\}$ de trouver une sous-famille de taille minimum couvrant $C = \cup_{i\in\{1,\ldots,n\}} S_i$. Il est très naturel de considérer ce problème comme la sélection d'éléments de $\mathcal{S}$, donc de choisir comme variable un vecteur caractéristique $\vec{x}$ de dimension $n = |\mathcal{S}|$, la formulation naturelle qui en découle est:

$$\mathrm{SC} = \left\{\begin{array}{ll} \min & \vec{1}_{|\mathcal{S}|} \cdot \vec{x} \\ & A\vec{x} \geq \vec{1}_{|C|} \\ & x_i \in \{0,1\} \end{array}\right.$$

où A est la matrice éléments-ensembles de l'instance, c'est à dire qu'un "1" dans la ligne $i \in \{1,\ldots,|C|\}$ et la colonne $j \in \{1,\ldots,|\mathcal{S}|\}$ signifie que l'élément i appartient à l'ensemble j. Dans cette formulation, le nombre d'ensembles (c'est à dire $|\mathcal{S}|$) apparaît comme pire valeur, en fait comme la pire valeur imposée par la formulation. Cependant, il est également très naturel d'imposer qu'il n'y ait pas plus d'ensembles dans la solution que d'éléments dans C puisqu'on peut toujours, dans le cas contraire, améliorer cette solution en sélectionnant au hasard, pour chaque élément, l'un des ensembles de la solution le contenant et ne garder en fin de compte que les ensembles de la solution sélectionnés au moins une fois. Cette remarque nous pousse alors à rajouter la contrainte $\vec{1}_{|\mathcal{S}|} \cdot \vec{x} \leq |C|$, on obtient une nouvelle formulation induite par la notion de pire valeur qui devient égale, dans ce cas, à $\min\{|\mathcal{S}|,|C|\}$. Ce nouveau problème — puisque nous avons vu que modifier la pire valeur modifie le problème — est a priori plus difficile que le premier; nous discutons son approximabilité dans l'article [8].

2.2 Pire valeur et approximabilité

Cette dernière remarque suggère que modifier la valeur du pire, c'est à dire ajuster les contraintes du problème, influe sur la difficulté de ce dernier, disons plutôt sur sa nature du point de vue de l'approximation. Rappelons que l'approximation polynomiale restreint de façon naturelle son champ d'investigation au cadre des problèmes d'optimisation car cette démarche a non a non seulement besoin d'une valeur objective pour évaluer la qualité des solutions trouvées, mais aussi de la notion de solution réalisable exprimant les conditions imposées aux solutions approchées. Cette remarque donne un premier élément de réponse à la question soulevée dans le paragraphe 2.1 concernant les différences de point de vue entre la résolution exacte et la résolution approchée. Parmi les contraintes d'un problème formulé selon la définition 1, on peut en distinguer de deux types. Certaines contraintes sont directement issues du problème NP-complet, sous forme de décision, et représentent ce qu'on impose a priori aux solutions, ou, plus précisément, définissent l'univers par rapport auquel l'optimisation est envisagée. C'est notamment le statut des contraintes de stabilité dans le problème de stable maximum. Il est clair, ou du moins c'est le choix qui est habituellement fait en théorie de l'approximation polynomiale, que les solutions approchées seront soumises à ces contraintes. Mais d'autres contraintes peuvent également être imposées pour spécifier l'univers des solutions acceptables parmi lesquelles on s'autorise à chercher des solutions. En d'autres termes, de telles contraintes correspondent au "cahier des charges des algorithmes approchés". La définition d'une solution approchée comme simple solution réalisable peut choquer, mais elle devient tout à fait compréhensible dès qu'on s'offre cette possibilité d'exprimer dans les contraintes même du problème les prérequis imposés à tout algorithme approché. C'est la philosophie même du cadre optimisation qui a été choisi pour l'approximation polynomiale: *tout dans la formulation.* La définition de la version optimisation d'un problème NP-complet contient donc plus d'informations que le problème originel. Elle spécifie non seulement l'objectif de la recherche exacte mais définit aussi les conditions d'une recherche approchée. Ainsi, formuler un problème selon la définition 1, c'est déjà envisager son approximation, ce qui est somme toute assez logique puisque le cadre optimisation a justement été introduit en vue de donner un sens à la recherche de solutions approchées.

Vu sous cet angle, la valeur optimale est déduite des contraintes du premier type tandis que la pire valeur serait plus le reflet des autres contraintes. L'inclure dans une mesure d'approximation correspond en quelques sortes à mesurer la qualité des algorithmes par rapport aux algorithmes acceptables et non par rapport à tout algorithme. Ajouter des contraintes ayant pour effet d'améliorer la pire valeur au sens de la fonction objective (ce que nous avons par exemple fait sur le problème de couverture d'ensemble ou encore sur le problème de stabilité en ajoutant la condition de maximalité) consiste à restreindre le champs des solutions acceptables, donc bien à rendre l'approximation plus difficile sans modifier pour autant le problème de recherche d'une solution exacte.

Le développement qui suit précise en quoi la modification des contraintes, même lorsqu'elle n'affecte pas la solution optimale, influe sur l'approximabilité du problème. Considérons deux problèmes Π_1 et Π_2 dont les instances courantes s'écrivent respectivement

$$I_1 = \begin{cases} \text{opt} & f(x) \\ & x \in \mathcal{C}_1 \end{cases} \qquad I_2 = \begin{cases} \text{opt} & f(x) \\ & x \in \mathcal{C}_2 \end{cases}$$

Nous supposons que Π_2 est déduit de Π_1 en ajoutant des contraintes (nous parlerons aussi de *troncature*), c'est à dire que toute instance I_2 est associée à une instance I_1 de telle sorte que $\mathcal{C}_2 \subseteq \mathcal{C}_1$ et que toutes les solutions optimales de I_1 sont dans $\mathcal{C}_2$ pour toutes instances couplées I_1, I_2 de Π_1 et Π_2 respectivement.

Ces deux problèmes sont les mêmes du point de vue de la recherche d'une solution exacte puisqu'ils ont les mêmes solutions optimales. Par contre, il semble assez logique, comme ils n'ont pas les mêmes solutions réalisables, donc pas les mêmes candidats à être solution approchée, qu'ils soient différents du point de vue de l'approximation. Tronquer les contraintes différencie donc radicalement les deux approches - exacte et approchée. La troncature "tue" certaines solutions réalisables. Par ailleurs, elle ne conserve les solutions exactes que dans la mesure où elle n'affecte pas les solutions optimales, ce qui suppose qu'on ait une idée de ces dernières. Deux situations peuvent alors se présenter.

Premier cas. Il peut être tout d'abord difficile, à partir d'une solution de I_1, de déduire une solution au moins aussi bonne dans $\mathcal{C}_2$. L'approximabilité de Π_2 apparaît alors comme un problème plus difficile que celle de Π_1.

Nous définissons dans la prochaine section des problèmes dérivés d'un problème d'optimisation NP-complet en ajoutant à chaque instance une information sur la valeur optimale. Cette condition peut, puisqu'elle est disponible dans l'instance, être intégrée dans les contraintes pour donner lieu à un nouveau problème qui aura les mêmes solutions optimales mais qui ne sera en général plus NP car la détermination d'une solution réalisable ne sera plus polynomiale. Dans ce cas, même si le problème originel est approximable, le problème résultant n'admettra aucun algorithme approché polynomial vu qu'un tel algorithme se doit de garantir la réalisabilité de la solution fournie.

Second cas. Etant donné un problème Π_1, considérons, comme précédemment, un problème Π_2 déduit de Π_1 en ajoutant des contraintes. Une situation très courante entre deux tels problèmes est le cas où toute solution d'une instance I_1 de Π_1 peut être polynomialement améliorée en une solution de l'instance correspondante I_2 de Π_2. Contrairement à l'exemple précédent, tout algorithme approché pour l'un des deux problèmes donne un algorithme approché pour l'autre avec au moins la même valeur. Mais pourquoi cette solution approchée commune serait-elle jugée de la même façon pour chaque problème? De par sa structure, le problème Π_2 impose à toutes ses solutions réalisables une valeur déjà relativement bonne pour Π_1. On peut même améliorer une solution pour Π_1 en une solution pour Π_2, mais ce qui apparaît comme un "luxe" pour le premier problème est une nécessité pour le second; la mesure d'approximation doit-elle oublier cela? Prenons par exemple, pour Π_1, le problème de stable maximum S et pour Π_2, le problème S′ pour lequel on impose aux ensembles stables d'être maximaux pour l'inclusion. Il est bien sûr toujours possible, à partir d'un ensemble stable donné, de trouver en temps polynomial ($O(n^2)$) un ensemble stable maximal pour l'inclusion le contenant.

Considérons, pour le problèmes S, l'algorithme glouton (algorithme 1) qui garantit un rapport d'approximation[2] $1/(\Delta + 1)$ où Δ est le degré maximum du graphe G.

ALGORITHME 1.

input: un graphe $G = (V, E)$; $\Gamma(x)$ est l'ensemble des voisins de x;
output: une partie P de V stable dans G;
begin
    $P \leftarrow \emptyset$;
    repeat
        sélectionner un sommet x de G de degré minimum;
        $P \leftarrow P \cup \{x\}$
        $G \leftarrow$ graphe engendré par $V \setminus \{\{x\} \cup \Gamma(x)\}$
    until $V = \emptyset$
end.

Cet algorithme s'applique directement à S′, cependant, pour certaines instances, il peut donner le plus petit stable maximal possible. Il suffit, par exemple, de considérer la suite d'instances G_n suivante: $G_n = (V_n, E_n)$ avec $V_n = V_n^1 \cup V_n^2 \cup V_n^3$, $|V_n^1| = n$, $|V_n^2| = 2n$,

[2]Ici, les deux notions de rapport (classique et différentiel) coïncident.

$|V_n^3| = n+1$; E_n est tel que V_n^3 soit une clique, les autres arêtes étant celles du graphe biparti complet entre V_n^2 et $V_n^1 \cup V_n^3$. L'algorithme 1, appliqué à G_n, fournit le stable V_n^1 qui se trouve être le plus petit stable maximal. Cet algorithme, assez bon pour S, doit-il être considéré aussi comme aussi bon pour S' alors qu'il peut, pour ce problème, donner la pire des solutions?

3 VALEUR OPTIMALE D'UN PROBLEME

Autant l'utilisation de la pire valeur dans l'évaluation de la qualité d'un algorithme approché est une approche nouvelle ou au moins restait jusqu'à présent marginalisée et très peu justifiée ([3, 15,16]), autant, par contre, la référence à la meilleure valeur fait-elle l'unanimité. Si la raison en semble tout à fait évidente, il est néanmoins intéressant de s'y arrêter quelques instants, et pour ce faire, il est judicieux de revenir sur les différentes problématiques de la théorie de la complexité.

3.1 Les trois problématiques de la théorie de la complexité

(i) *La problématique de la décision*: une instance se présente comme une question, et le but est d'y répondre par *oui* ou *non*.

En fait, les problèmes NP-complets originels sont tous définis sous cette problématique ([13]). Un algorithme susceptible de résoudre un problème de décision sera appelé *algorithme décisionnel.*

(ii) *La problématique de l'optimisation*: une instance se présente comme un programme d'optimisation du type

$$\begin{cases} \text{opt} & f(x) \\ & x \in C \end{cases}$$

où opt $\in \{\min, \max\}$, f est la fonction objective et C l'ensemble des solutions réalisables. Le problème consiste alors à déterminer une solution $\hat{x}$ optimale, et un algorithme répondant à cette exigence sera qualifié de *constructif.*

Le champs d'investigation de cette problématique est plus restreint que celui de la première puisqu'il est bien connu qu'à tout problème d'optimisation correspond un problème de décision alors que seulement certains problèmes de décision admettent une version optimisation. Deux raisons peuvent expliquer l'intérêt accordé à cette problématique: du point de vue des applications c'est la plupart du temps elle qui s'impose, mais surtout, comme nous l'avons déjà mentionné, elle offre la possibilité, par rapport à la problématique de la décision, de définir une notion d'approximation. Ainsi, c'est en son sein que s'est développée la *problématique de l'optimisation approchée* (**(ii)-ap**) qui consiste, étant donné un problème d'optimisation, à déterminer une solution réalisable offrant certaines garanties de qualité. Un algorithme répondant à cette problématique sera appelé *algorithme approché constructif.*

(iii) *La problématique de la valeur optimale*: les instances sont les mêmes que pour la problématique de l'optimisation, cependant l'objectif est cette fois de déterminer la valeur optimale; un algorithme correspondant sera alors qualifié de *non constructif.* On peut, ce qui est toutefois moins courant, définir dans ce cadre une notion d'approximation (**(iii)-ap**), un *algorithme approché non constructif* détermine une valeur réalisable en garantissant une certaine qualité par rapport à la valeur optimale. En fait, si on adopte la problématique de la décision pour laquelle la question porte sur la *valeur* de la solution optimale et non sur la solution elle-même, le cadre **(iii)-ap** apparaît comme le plus naturel pour développer la notion d'approximation. Cependant, la plupart des algorithmes déterminant la valeur d'un problème étant constructifs, les recherches dans le domaine se focalisent plutôt sur la problématique de l'optimisation.

La valeur optimale s'impose comme l'une des quantités les plus pertinentes pour tester la qualité d'un algorithme approché, qu'il soit constructif ou pas. Par ailleurs, un problème de décision, lorsqu'une version optimisation existe, revient toujours à comparer la valeur optimale de la version optimisation à une constante donnée dans l'instance. La notion de valeur optimale joue donc un rôle charnière entre les trois problématiques enoncées précédemment.

Aussi naturelle que puisse être cette valeur, il est très intéressant de réfléchir à ses liens avec la difficulté du problème, comme nous l'avons fait à propos de la pire valeur. Cette question revient à étudier les liens entre ces trois problématiques et plus particulièrement, si on se place dans le cadre de l'approximation, entre les problématiques **(ii)-ap** et **(iii)-ap**.

3.2 Valeur optimale et approximabilité

La première remarque naïve est qu'un algorithme constructif est en particulier non constructif et que ce dernier est également décisionnel. Ainsi, *la problématique de l'optimisation est au moins aussi difficile que la problématique de la valeur optimale, elle même au moins aussi difficile que la problématique de la décision.* De même, un algorithme approché constructif est approché non constructif; ce dernier n'est par contre, en général, pas décisionnel; si tel est le cas, le problème de l'approximation est alors lui même NP-complet. *L'approximation constructive est donc au moins aussi difficile que l'approximation non constructive.* La question naturelle qui se pose est alors, *en quel sens l'optimisation (resp. l'approximation) constructive est-elle plus difficile que l'optimisation (resp. l'approximation) non constructive*, ou encore *en quoi une connaissance sur la valeur d'un problème aide-t-elle à en déterminer une solution (resp. solution approchée)?*

Nous proposons deux points de vue différents pour aborder cette question en fonction, en quelque sorte, de la portée de l'information supposée sur la valeur, c'est à dire selon qu'elle porte sur le problème ou sur une instance particulière.

Premier point de vue. En vue de représenter une information portant, de façon globale, sur la valeur optimale de toutes les instances d'un problème, il est utile d'introduire la notion d'*oracle*. Il s'agit d'une application ayant pour domaine l'ensemble des instances d'un problème (par exemple, dans le développement qui suit, l'oracle fournit, pour une instance, sa valeur optimale). Le terme *oracle* fait référence au fait que cette application est une "boite noire", c'est à dire, d'une part, qu'on s'autorise l'accès, pour une instance quelconque, au résultat de cette application sans expliciter son fonctionnement interne et que, d'autre part, l'appel à l'oracle se fait en temps constant[3]. Formellement, cet artifice correspond à introduire une composante non déterministe dans l'algorithme. Remarquons alors qu'il suffit de substituer une procédure (déterministe) à l'oracle pour rendre l'algorithme originel déterministe. Dans ce cas, il faut bien entendu tenir compte, dans l'évaluation de la complexité, du nombre d'appels à l'oracle et de la complexité de cette procédure.

La première question qu'il est naturel de poser est **Q1**: *étant donné un problème d'optimisation* Π*, existe-t-il un algorithme constructif polynomial pour* Π *utilisant un oracle donnant, pour toute instance, sa valeur?*

Remarquons que dans ce cas, s'il existe un algorithme polynomial non constructif, il suffit de l'utiliser comme oracle pour concevoir un algorithme polynomial constructif.

La réponse à cette question est positive dans le cas où la version décision de Π est NP-complète ([4]). En d'autres termes, dans le cas où la problématique de la décision est NP-complète, les deux problématiques, de la valeur et de l'optimisation, prises dans leur globalité, sont de difficulté équivalente.

Pour illustrer ce résultat, considérons par exemple le problème de stable maximum, l'algorithme 2 utilisant l'oracle noté **VAL** répond à la question.

[3] Ce qui revient à ne pas tenir compte de ces appel dans l'évaluation de la complexité

ALGORITHME 2.

input: un graphe $G = (V, E)$;
output: un stable S maximum pour G;
begin

$S \leftarrow \emptyset$;
while $V \neq \emptyset$ **do**
  choisir un sommet $v \in V$;
  $V_c \leftarrow V \setminus \{v\}$;
  $E_c \leftarrow E \setminus \{vv', vv' \in E\}$;
  $G_c \leftarrow (V_c, E_c)$
  if $\mathbf{VAL}(G) \neq \mathbf{VAL}(G_c)$ **then**
    $S \leftarrow S \cup \{v\}$;
    $V \leftarrow V \setminus (\{v\} \cup \Gamma(v))$;
    $E \leftarrow \setminus\{ij, ij \in E \wedge i \notin V\}$;
    $G \leftarrow (V, E)$
  else $G \leftarrow G_c$
od

end.

Par contre, le problème de l'approximation sous ce point de vue n'a, à notre connaissance que très peu été abordé. Pour le problème de stable maximum d'un graphe, par exemple, une question ouverte intéressante serait: *en quoi l'accès à un oracle* **APR** *capable de fournir, pour un graphe quelconque, la cardinalité A d'un stable vérifiant $A \geq \rho\alpha(G)$, avec $0 < \rho < 1$, permettrait-il de construire un stable garantissant un rapport d'approximation constant?* Dans le cas de l'approximation, cette question semble beaucoup plus difficile à aborder que dans le cas de la résolution exacte. La seule remarque que nous puissions faire pour l'instant, mais qui reste dans le cadre non-constructif, est qu'on peut utiliser l'oracle **APR** pour déterminer un schéma d'approximation polynomial non constructif[4]:

PROPOSITION 1. *Pour tout $\epsilon \in]0,1[$, il existe un algorithme polynomial utilisant l'oracle* **APR** *fournissant la cardinalité $\lambda(\epsilon)$ d'un stable tel que $\lambda \geq (1-\epsilon)\alpha(G)$.*

PREUVE: Fixons $\epsilon > 0$ et déterminons le plus petit entier m tel que $\rho^{1/m} \geq (1-\epsilon)$. Considérons alors le graphe G^m défini par la récurrence suivante:

$$\begin{aligned} G_1 &= G \\ G^i &= G \times G^{i-1} \end{aligned}$$

où l'opération $G_1 \times G_2$ signifie la composition (définie dans [5]). Notons $\alpha(G)$ (resp. $\alpha(G^m)$) le nombre de stabilité de G (resp. G^m); il est prouvé que (cf. [5]), chaque stable de cardinalité $\tilde{\alpha}(G)$ de G peut-être transformé en temps polynomial en un stable de cardinalité $\tilde{\alpha}_m(G^m) = \tilde{\alpha}(G)^m$ dans G^m et vice-versa. Par conséquent $\alpha(G^m) = \alpha(G)^m$. Ainsi, en construisant G^m et en interrogeant l'oracle **APR** sur ce dernier graphe, on obtient la cardinalité A_m d'un stable de G^m vérifiant $A_m \geq \rho\alpha(G^m)$. Mais alors, $A = (A_m)^{1/m}$ est la cardinalité d'un stable de G vérifiant $A \geq \rho^{1/m}\alpha(G) \geq (1-\epsilon)\alpha(G)$. ∎

Second point de vue. Notons que l'algorithme 2 nécessite $O(n)$ appels différents à l'oracle qui doit donc admettre un graphe quelconque en entrée. Que se passe-t-il par contre si l'information sur la valeur est attachée à l'instance considérée du problème? Supposons des problèmes d'optimisation pour lesquels est incluse dans l'instance même une information

[4] Une suite d'algorithmes polynomiaux non constructifs, indicée par ϵ, garantissant, pour chaque ϵ, un rapport d'approximation $1-\epsilon$.

sur la valeur optimale, sans que ces instances soient nécessairement reconnaissables en temps polynomial. Notons encore, comme dans la section précédente, que ces problèmes sortent alors légèrement du cadre habituel[5]. Considérons par exemple, comme information sur la valeur optimale, la donnée de cette valeur elle-même[6]. Ainsi, pour un problème d'optimisation Π on définit le problème $\tilde{\Pi}$ dont une instance $\tilde{I}$ est un couple $(I, \beta(I))$ où I est instance de Π et $\beta(I)$ sa valeur optimale. Pour $\tilde{\Pi}$, la problématique de la valeur est bien sûr polynomiale, puisqu'il suffit de lire la réponse dans l'instance elle-même. La problématique de l'optimisation consiste alors à déterminer, à partir du couple $(I, \beta(I))$, une solution optimale pour I. Remarquons à ce propos que I et $\tilde{I}$ ont les mêmes solutions réalisables (resp. optimales) et donc, que la reconnaissance de la réalisabilité pour $\tilde{I}$ est polynomiale dès qu'elle l'est pour I. Analyser la difficulté relative des problématiques **(iii)** et **(ii)** (resp. **(ii)-ap**) pour Π équivaut à étudier la difficulté (question **Q2** (resp. **Q2-ap**)) de la problématique **(ii)** (resp. **(ii)-ap**) pour $\tilde{\Pi}$.

A titre d'exemple, posons le problème ouvert suivant, qui est un cas particulier, respectivement, des questions **Q2** et **Q2-ap**: *étant donné un problème d'optimisation restreint aux instances ayant même valeur que leur relaxé, où le programme relaxé de l'instance*

$$\begin{cases} \text{opt} & \vec{x} \\ & \vec{x} \in C \\ & x_i \in \{0,1\} \end{cases}$$

est le programme

$$\begin{cases} \text{opt} & \vec{x} \\ & \vec{x} \in C \\ & x_i \geq 0 \end{cases}$$

quelle est la difficulté de la problématique **(ii)** *(resp.* **(ii)-ap***)?*

Pour de tels problèmes, la programmation linéaire étant polynomiale, ([2]), la valeur optimale d'une instance donnée est calculable en temps polynomial; mais qu'en est-il de la détermination d'une solution optimale?

Commençons par considérer, à titre d'exemple, le cas du problème de stable maximum d'un graphe. Les instances, formulées par le programme linéaire S du paragraphe 2.1, qui ont même valeur que leur version relaxée (contrainte $\vec{x} \in \{0,1\}^{|V|}$ relaxée en $\vec{x} \geq \vec{0}_{|V|}$) correspondent exactement ([6,11]) aux graphes König-Egervary définis par l'égalité $\alpha(G) = \beta(G)$ où $\alpha(G)$ désigne le nombre de stabilité du graphe G et $\beta(G)$ la cardinalité d'une couverture minimum d'arêtes. Pour cette classe de graphes, la construction d'un stable maximum est connue comme polynomiale ([12]). Par conséquent, dans cet exemple, l'information sur la valeur que nous avons incluse a priori dans l'instance est trop forte pour mettre en évidence la limite entre les points de vue constructif et non constructif.

Plus généralement, le résultat suivant permet de justifier qu'il est très probable que la connaissance de la valeur optimale d'une instance n'apporte, la plupart du temps, aucune aide pour la détermination d'une solution optimale (resp. ρ-approchée). Notons néanmoins que, si la connaissance de la valeur optimale d'une instance rendait la résolution constructive de cette instance polynomiale, la réponse à la question **Q1** apparaîtrait comme corollaire.

THEOREME 1. *Soit Π un problème d'optimisation tel qu'il existe une borne polynomiale M vérifiant, pour toute instance I de Π, $|\beta(I)| \leq M(I)$. S'il existe un algorithme polynomial constructif (repectivement constructif ρ-approché) $\mathcal{A}$ de complexité exacte $I \mapsto f(I)$ évaluable en temps polynomial pour $\tilde{\Pi}$, alors il en est de même pour Π.*

[5]Ils ne sont en particulier pas NPO au sens de l'article [4].

[6]La problématique **(iii)-ap** est alors écartée.

PREUVE: Supposons qu'un tel algorithme $\mathcal{A}$ existe, il suffit, pour $k \in [-M(I), M(I)]$, d'appliquer l'algorithme $\mathcal{A}$ pendant $f(I)$ étapes élémentaires sur les données (I, k). Pour chaque k, ce procédé mène soit à une impossibilité, soit à une solution non réalisable (auquels cas $k \neq \beta(I)$), soit enfin à une solution réalisable (en particulier lorsque $k = \beta(I)$). Il suffit alors de choisir, parmi les solutions réalisables obtenues, celle de meilleure valeur objective. ∎

Remarquons avant tout que la contrainte d'existence de la borne M est vérifiée pour la plupart des problèmes connus; à titre d'exemple, pour le problème de stable maximum, il suffit de prendre pour M l'ordre (nombre de sommets) du graphe instance.

Le problème de stabilité S_κ. Considérons maintenant, à partir du problème de stable maximum, la transformation consistant à intégrer dans l'instance une information plus lâche sur la valeur optimale, à savoir seulement son ordre de grandeur par rapport à la taille n du graphe instance. Cette démarche nous conduit plus précisément aux problèmes suivants que nous avons introduits dans l'article [10]. Pour toute constante $\kappa > 1$, on définit le problème de stabilité S_κ correspondant à la restriction du problème de stable maximum aux instances (graphes d'ordre n) dont le nombre de stabilité est supérieur à n/κ. On définit également le problème S_{κ^∞} dont une instance courante est un couple (G, κ) où $\kappa \geq 1$ et G est un graphe d'ordre (nombre de sommets) n dont le nombre de stabilité $\alpha(G)$, vérifie $\alpha(G) \geq n/\kappa$. L'objectif est, comme pour le problème général, de déterminer un stable maximum de G.
Ces problèmes sont particulièrement intéressants à envisager aussi bien dans le cadre de cette réflexion sur les valeurs extrémales que sur la difficulté respective des problèmes d'évaluation de valeur est de construction de solutions. La première remarque concernant ce type de problèmes est que la reconnaissance de leurs instances est NP-complete dès que $\kappa \geq 3$:

PROPOSITION 2. *$\forall \kappa \geq 3$, $\kappa \in \mathbb{N}$, le problème de décider si un graphe G d'ordre n vérifie $\alpha(G) \geq n/\kappa$ est NP-complet.*

PREUVE: Pour tout graphe $G = (V, E)$, considérons le graphe ${}^\kappa G = ({}^\kappa V, {}^\kappa E)$ défini comme suit: ${}^\kappa V = V \times \{1, \dots, \kappa\}$ et $\forall((v,i),(v',j)) \in {}^\kappa V \times {}^\kappa V$, $((v,i),(v',j)) \in {}^\kappa E \Leftrightarrow ((i=j) \wedge (v,v') \in E \vee (i \neq j) \wedge (v = v'))$. Il est clair que ${}^\kappa G$ est d'ordre κn où n est l'ordre de G et $\alpha({}^\kappa G) \leq n$. Plus précisément, tout stable d'ordre λ de ${}^\kappa G$ correspond, dans G à une coloration de λ sommets en au plus κ couleurs, et réciproquement. Ainsi, $\alpha({}^\kappa G) \geq |{}^\kappa V|/\kappa \Leftrightarrow \alpha({}^\kappa G) = n \Leftrightarrow G$ est κ-colorable. Or, pour tout entier $\kappa \geq 3$, la reconnaissance de la κ-colorabilité est NP-complet [13], d'où le résultat. ∎

Ce résultat justifie que l'information sur la valeur optimale intégrée dans les instances de ces problèmes est très forte, aussi est-il naturel de se de demander si cette information est exploitable pour l'approximation de ces problèmes. On a alors le résultat suivant.

THEOREME 2. *Pour toute constante ρ, aucun algorithme polynomial (constructif) pour S_{κ^∞} ne peut garantir un rapport supérieur ou égal à ρ.*

Autrement dit, S_{κ^∞} n'admet pas d'algorithme à rapport constant. Pour les problèmes S_κ, ce résultat signifie que s'il existe un algorithme polynomial garantissant, pour tout problème S_κ, un rapport d'approximation $\rho(\kappa)$, alors la fonction $\kappa \mapsto \rho(\kappa)$ tend vers zéro lorsque κ tend vers l'infini. En fait, on peut même préciser ce résultat en évaluant la vitesse de convergence de ρ vers zéro:

THEOREME 3. *Il existe $\varepsilon > 0$ et κ_0 tel que $\forall \kappa \geq \kappa_0$, il n'existe pas d'algorithme polynomial en l'ordre du graphe (mais éventuellement exponentiel en κ) garantissant le rapport d'approximation $(1/\kappa)^\varepsilon$*

PREUVE: Dans l'article [1] il est prouvé que le problème du stable, même pour les graphes à degré maximum borné (supérieurement par une constante fixée et indépendante de n; notons cette classe des problèmes du stable par B(Δ)S) n'admet pas de schéma d'approximation polynomial[7] (PTAS). En utilisant ce résultat, nous prouvons le lemme suivant (lemme 1).

LEMME 1. *Il existe κ_0 tel que S_{κ_0} n'admet pas de PTAS.*

PREUVE: Prouvons que si, $\forall\kappa$, S_κ admet un PTAS, alors B(Δ)S admet aussi un PTAS.

En effet, dans un un graphe G d'ordre n, chaque stable maximal est supérieur ou égal à $n/(\Delta+1)$, par conséquent $\alpha(G) \geq n/(\Delta+1)$ et, pour chaque Δ, B(Δ)S est un sous-problème de $S_{\Delta+1}$, résolu par un PTAS sur l'hypothèse que $\forall\kappa$, S_κ admet un PTAS, c.q.f.d. ∎ **(Lemme 1.)**

Soit alors G une instance de S_{κ_0} et d'ordre n. Pour tout entier m, considérons le graphe G^m défini dans la preuve de la proposition 1. Si n est l'ordre de G et n_m celui de G^m, alors $n_m = n^m$. Les propriétés de cette construction permettent d'établir que, si G est instance de S_κ, alors le graphe G^m est une instance du problème S_{κ^m}.

Supposons alors que, $\forall\varepsilon > 0, \exists\kappa > \kappa_0$ tel que S_κ est $(1/\kappa)^\varepsilon$-approximable. Pour $\alpha > 0$, choisissons $\varepsilon > 0$ tel que $(1/\kappa_0^{3/2})^\varepsilon > 1-\alpha$ et $\kappa > kappa_0$ tel que S_κ soit $(1/\kappa)^\varepsilon$-approximable. On détermine alors l'entier m tel que $\kappa_0^m \leq \kappa < \kappa_0^{m+1}$, le graphe G^m est instance de S_κ, si bien que l'algorithme supposé pour ce dernier problème permet de déterminer en temps polynomial un stable de cardinalité $\alpha'(G^m) \geq (1/\kappa)^\varepsilon\alpha(G^m) \geq (1/\kappa_0^{1+m})^\varepsilon\alpha(G^m)$. D'après la discussion qui précède, on peut en déduire un stable de G de cardinalité $\alpha'(G^m)^{1/m} \geq (1/\kappa_0^{(1+m)/m})^\varepsilon(\alpha(G^m))^{1/m} \geq (1/\kappa_0^{3/2})^\varepsilon\alpha(G) \geq (1-\alpha)\alpha(G)$. Ceci étant fait pour tout $\alpha \in]0,1[$, on a déterminé un schéma d'approximation polynomial pour S_κ qui constitue une contradiction. ∎

Par contre, nous ne sommes pas encore en mesure de caractériser la difficulté de l'approximation des problèmes S_κ à κ fixé. La seule remarque à ce propos est que pour $\kappa > 2$, S_κ est approximable à rapport constant. En effet, considérons une instance G de $S_{2-\epsilon}$ pour une constante positive ϵ. Alors, $\alpha(G) \geq n/(2-\epsilon) \geq [(1/2)+\epsilon']n$ pour $\epsilon' > 0$. Notons par e le nombre des sommets exposés[8] par rapport à un couplage maximum M de cardinalité m. On a $n = 2m+\epsilon$ et $\alpha(G) \leq m+\epsilon$. Si nous choisissons comme solution approchée pour le stable l'ensemble des sommets exposés, la combinaison des inégalités précédentes donne $e \geq 2\epsilon' n$. Le rapport constant $2\epsilon'$ est est donc assuré pour pour $S_{2-\epsilon}$. Pour le cas $\kappa \geq 2$, la question de l'approximabilité est particulièrement intéressante, nous en discutons certains enjeux dans l'article [10].

Un problème légèrement différent de l'approximabilité à rapport constant de S_κ est le problème P_κ suivant: *étant donné une instance G de S_κ, déterminer un stable de taille n/κ.* Il s'agit d'un problème purement constructif plus difficile que l'approximation à rapport $1/\kappa$ de S_κ. Pour ce problème, on déduit par contre immédiatement de la proposition 2 le résultat suivant:

PROPOSITION 3. *$\forall\kappa \geq 3$, $\kappa \in \mathbb{N}$, il ne peut exister de couple d'algorithmes polynomiaux en n (éventuellement exponentiels en κ) $(\mathcal{A}, f)$ où, pour tout graphe G d'ordre n et instance de S_κ, $\mathcal{A}$ détermine un stable de taille supérieure à n/κ avec une complexité inférieure à $f(n)$.*

La preuve est simple en conjugant les idées de la preuve du théorème 1 au résultat de la proposition 2.

4 REMARQUES FINALES

Ces éléments de réflexion sur les notions de valeurs (et solutions) optimale et pire et leurs liens avec l'approximabilité des problèmes justifient a posteriori de mesurer la qualité d'un

[7] Une suite d'algorithmes polynomiaux, indicés par ϵ, garantissant, pour chaque ϵ, un rapport d'approximation $1-\epsilon$.

[8] Qui est non vide à cause de l'hypothèse $\alpha(G) > n/2$.

algorithme approché à partir de ces valeurs. Ces arguments complètent donc ceux déjà avancés dans l'article [9] et donnent quelques pistes vers l'intégration du choix de ces paramètres dans l'approche axiomatique qui nous a conduits à proposer le rapport d'approximation différentiel.

Cette discussion nous a également menés à des considérations concernant les liens entre les différentes problématiques de la complexité et plus précisément entre les notions d'algorithmes constructifs et non constructifs. Ces liens nous semblent une question clef de la recherche actuelle en théorie de la complexité et plus particulièrement de l'approximation polynomiale. Ils focalisent actuellement l'attention de nombreux chercheurs si on en juge par plusieurs publications récentes sur le sujet (citer ces travaux nous conduirait trop loin, mais pour obtenir plus d'informations, on pourra en particulier se référer à la référence [4] qui est un survey très complet). Très peu de résultats permettent encore de spécifier ces liens dans le cadre de l'approximation alors que du point de vue de la résolution exacte, ils sont beaucoup mieux connus. En effet, du point de vue approximation, tous les résultats positifs actuels sont constructifs alors que tous les résultats négatifs, notamment sur les problèmes de stable maximum ([1]), de coloration et couverture d'ensembles ([14]) sont non constructifs. Par conséquent, aucun de ces résultats ne permet encore de mettre en évidence une différence de difficulté entre l'approximation constructive et non constructive.

La théorie de la complexité et l'approximation polynomiale sont à la frontière entre des enjeux appliqués et des enjeux théoriques, notamment en logique. Les premiers sont généralement liés au point de vue constructif tandis que les seconds s'inscrivent plutôt dans le cadre non constructif. L'étude des liens entre ces deux problématiques serait très intéressant en vue d'unifier ces différents aspects de l'algorithmique moderne. En particulier, la discussion engagée dans cet article sur la classe des problèmes S_κ pourrait founir une première piste pour distinguer ces deux approches.

Comme les notion d'équivalence entre problèmes ([9]), ce type de considérations différencie radicalement l'approximation polynomiale de la théorie classique de la complexité. Ce phénomène est une illustration du fait que la théorie originelle de la complexité est définie en termes non constructif - décisionnel alors que l'approximation polynomiale l'est plutôt en termes constructifs. Ceci provient d'une part de ce que la notion d'approximation nécessite un cadre d'optimisation et d'autre part de ce qu'elle est fondée, à l'origine, sur des considérations pragmatiques et appliquées ayant pour but principal la résolution effective de problèmes NP-complets. Ainsi, son cadre a été "greffé" à l'origine sur la théorie classique de la complexité avec quelques ambiguïtés, mais il suffisait pour répondre pleinement au premier objet de l'approximation, à savoir la conception de résultats positifs. En effet les résultats constructifs sur lesquels sont fondés tous les résultats positifs actuels sont, d'une part, plus forts que des résultats non constructifs et répondent parfaitement, d'autre part, aux exigences des applications. Ils sont donc de ce point de vue tout à fait satisfaisants.

Cependant, comme comme la plupart des théories mathématiques issues d'enjeux pragmatiques, l'approximation a très vite dépassé ce cadre d'origine et s'est révélée être un outil théorique remarquable pour la compréhension de la classe NP et la mise en évidence d'une structure en son sein. C'est à ce niveau que la distinction entre constructif et non constructif apparaît, et l'étude de cette question est liée de façon naturelle au problème de définir directement un cadre formel à cette théorie. Ceci reviendrait à définir, indépendamment des notions classiques de complexité, une complexité du point de vue constructif pour des problèmes d'optimisation.

BIBLIOGRAPHIE

[1] ARORA S., LUND C., MOTWANI R., SUDAN M., SZEGEDY M., "proof verification and intractability of approximation problems", *Proc. FOCS* (1992), 14-23.

[2] ASPVALL B., STONE R. E., "Khachiyan's linear programming algorithm", *J. Algorithms*, 1 (1980), 1-13.

[3] AUSIELLO G., D'ATRI A., PROTASI M., "structure preserving reductions among convex optimization problems", *J. Comput. System Sci.*, 21 (1980), 136-153.

[4] AUSIELLO G., CRESCENZI P., PROTASI M., "approximate solutions of NP optimization problems" *Technical Report* SI/RR-94/03 (1994), Università di Roma "La Sapienza".

[5] BERGE C., *graphs and hypergraphs*, Amsterdam, North Holland, 1973.

[6] BOURJOLLY J. - M., HAMMER P. L., SIMEONE B., "node-weighted graphs having the König-Egervary property", Math. Prog. Study, 22 (1984), 44-63.

[7] DEMANGE M., *approximation polynomiale de problèmes NP-complets et programmation linéaire: une nouvelle mesure d'approximation et algorithmes*, thèse de Doctorat, Université Paris I, 1994.

[8] DEMANGE M., GRISONI P., PASCHOS V. TH., "differential approximation algorithms for some combinatorial optimization problems", *Cahier Eco&Maths*, 94.65 (1994), Université Paris I.

[9] DEMANGE M., PASCHOS V. TH., "on an approximation measure founded on the links between optimization and polynomial approximation theory", *Theoretical Comp. Sci.*, à à paraître.

[10] DEMANGE M., PASCHOS V. TH., "the approximability behaviour of some combinatorial problems with respect to the approximability of a class of maximum independent set problems", *Computational Optimization and Applications*, à paraître.

[11] DEMANGE M., PASCHOS V. TH., "exact and approximation results on maximum independent set and minimum vertex covering - graphs with great stability number", *Cahier Eco&Maths*, 94.68 (1994), Université Paris I.

[12] R. W. DEMING, "Independence Numbers of Graphs - An Extension of the König-Egervary Theorem",*Discrete Maths* 27, pp. 23-33, 1979.

[13] GAREY M. R., JOHNSON D. S., *computers and intractability: a guide to the theory of NP-completeness*, San Francisco, Freeman and Company, 1979.

[14] LUND C., YANNAKAKIS M., "on the hardness of approximating minimization problems", *preprint AT&T Bell Laboratories* (1992).

[15] VAVASIS S. A., "approximation algorithms for indefinite quadratic programming", Math. Programming, 57 (1992), 279-311.

[16] ZEMEL E., "functions for measuring the quality of approximate solutions to zero-one programming problems" *Discussion Paper* (1978), Graduate School of Management, Northwestern University, Evanston, Illinois.