VERSIONS ON-LINE DU PROBLEME DU STABLE MAXIMUM

Bruno Escoffier,
Doctorant en informatique
escoffier@lamsade.dauphine.fr

Vangélis Th. Paschos,
Professeur en informatique
paschos@lamsade.dauphine.fr

Adresse professionnelle
Lamsade, Université Paris IX Dauphine ★ Place du maréchal de Lattre de Tassigny★
75775 Paris cedex 16

Résumé : Dans un problème on-line, l'instance du problème n'est pas entièrement connue au départ mais est "révélée" en plusieurs étapes, le but étant de construire, au cours de ce processus où l'on découvre l'instance, une solution réalisable la meilleure possible. Des modèles on-line pour le problème du stable ont été étudiés (voir notamment "*On-line maximum-order induced hereditary subgraph problem*" M. Demange, X. Paradon, V. Th. Paschos. SOFSEM 2000, p. 327-335) : l'instance y est alors initialement vide, et à chaque étape des sommets apparaissent jusqu'à l'obtention du graphe final. Nous allons ici étudier l'approximabilité de nouveaux modèles on-line : le premier modèle consistera à changer le mode de révélation de l'instance. Dans le deuxième, on étudiera le modèle classique mais en le relâchant par l'introduction d'une possibilité de backtracking. Cette étude sera faite sous l'angle de l'approximation : on recherchera ainsi des résultats positifs (algorithmes à garanties de performance) et des résultats négatifs (bornes sur les garanties possibles).

Summary : In an on-line problem, the instance is not completely known at the beginning, it's given in many stages. The goal is to build, during this process, a feasible solution as good as possible. An on-line model specifies how the instance is revealed and how an algorithm can construct its solution. Some on-line models for the max independent set problem have been studied (see "*On-line maximum-order induced hereditary subgraph problem*" M. Demange, X. Paradon, V. Th. Paschos. SOFSEM 2000, p. 327-335) : in such models, the instance is empty at the beginning. At each stage, a vertex (or a set of vertices) appears, and so on until the graph is completely revealed. Here, we will study approximability of new versions of the on-line independent set problem : in the first one, we change the way the instance is given; in the second one, we consider the 'classical' model but we introduce a relaxation : a possibility of backtracking. We will focus on finding approximation properties of such problems, i.e. both positive results (approximation algorithms) and negative ones (bounds on approximability).

Mots clés : optimisation combinatoire, algorithmique on-line, ensemble stable, algorithmes approchés, garanties de performance.

Versions on-line du problème du stable maximum

1 - INTRODUCTION

1.1 - Problème du stable et rapport d'approximation

Rappelons tout d'abord qu'un stable d'un graphe *G=(V,E)* (où *V* est l'ensemble des sommets et *E* l'ensemble des arêtes) est un sous-ensemble *S* de sommets deux à deux non adjacents, c'est-à-dire tel que: $(x,y) \in S^2 \Rightarrow (x,y) \notin E$. Le but du problème est alors, étant donné un graphe, de trouver un stable de ce graphe de cardinal maximal.

Le problème du stable maximum est NP-difficile, et nous allons nous placer dans le cadre de la recherche de résolutions approchées. Un algorithme approché *A* pour un problème d'optimisation fournit, pour toute instance du problème, une solution réalisable (ici un stable). On mesure alors la qualité de la solution fournie par *A* sur le graphe *G* en regardant le rapport $\frac{|A(G)|}{\alpha(G)}$, où $|A(G)|$ désigne le cardinal de la solution fournie par *A* et $\alpha(G)$ désigne le cardinal d'une solution optimale (que l'on nomme souvent nombre de stabilité du graphe).

On dit qu'un algorithme garantit *f(G)* (où *f* est une fonction de l'ensemble des instances *G* à valeurs dans [0,1]) si pour toute instance *G*, $\frac{|A(G)|}{\alpha(G)} \geq f(G)$. On s'intéresse bien sûr particulièrement à des garanties d'algorithmes polynomiaux, et souvent à des garanties fonctions uniquement de la taille de l'instance.

Pour le problème du stable, la meilleure garantie polynomiale connue (en fonction du nombre *n* de sommets du graphe instance *G*) est en $O\left(\frac{\log^2(n)}{n}\right)$ (voir [Boppana, 1992]).

Rappelons également qu'il existe une version pondérée du problème du stable : chaque sommet a un poids, le but étant alors de trouver un stable de poids maximal (le poids d'un sous-ensemble de sommets étant bien entendu défini comme la somme des poids des sommets). Du point de vue de l'approximation polynomiale, il est intéressant de noter que l'on dispose également d'un algorithme offrant une garantie en $O\left(\frac{\log^2(n)}{n}\right)$ (voir [Halldorsson, 2000a]).

1.2 - Problèmes on-line

L'algorithmique dite "on-line" est apparue pour modéliser des situations où l'instance d'un problème n'est pas entièrement connue quand commence la résolution effective. Il est par exemple fréquent que l'on doive commencer cette résolution dès réception des premières informations (problèmes en-ligne, avec contraintes de temps de réponse,…).

De nombreux problèmes d'optimisation combinatoire classiques ont été étudiés dans des versions on-line (on spécifie alors simplement des règles concernant la manière dont les instances sont révélées et la manière de construire une solution), voir par exemple [Ausiello,2001] pour le problème du voyageur de commerce, [Halldorsson, 2000b] et [Halldorsson, 1994] pour la coloration, [Demange, 2000], [Halldorsson, 2002], [Paradon, 2000] et [Paschos, 2001] pour le stable.

Considérons par exemple la situation suivante : une entreprise reçoit, régulièrement, des contrats pour des commandes de matériels. Cette entreprise, disposant de moyens (techniques, humains,...) limités, ne peut honorer tous les contrats, et est ainsi contrainte d'en refuser certains (son but étant d'en accepter le plus possible). Quand elle reçoit une proposition, l'entreprise doit, évidemment, donner sa réponse dans un intervalle de temps limité (soit très rapidement, soit à la fin du mois par exemple).

Modélisons cette situation par un graphe dont les sommets sont les contrats. Imaginons que les incompatibilités se résument à des paires de contrats que l'entreprise ne peut honorer simultanément, ces paires constituant les arêtes du graphe. Le problème de l'entreprise est alors de trouver un stable de taille maximale dans ce graphe.

Cet exemple permet de définir le modèle le plus naturel pour le stable on-line. Un modèle consiste à définir comment l'instance est révélée à l'algorithme, et quels sont les droits de celui-ci pour construire sa solution. Ici, nous considérons donc que le graphe initial est vide. A chaque itération, on a un ensemble de sommets qui se rajoutent au graphe courant (les nouveaux contrats dans l'exemple). Un algorithme doit alors choisir de manière irrévocable quels sommets il met dans sa solution (quels contrats il accepte). Ce modèle contient un cas particulier, celui où les sommets arrivent un par un (c'est le cas où l'on doit choisir dès réception du contrat).

1.3 - Résultats précédents et objectifs de l'étude

C'est le modèle précédent qui a été étudié dans [Demange, 2000]. L'analyse y est faite sous l'angle de la recherche d'algorithmes à garanties de performance (en étudiant toujours le rapport du cardinal de la solution fournie par un algorithme on-line par le nombre de stabilité du graphe). Il est à noter que, contrairement au problème classique (que l'on appellera désormais off-line), la recherche de garantie pour des algorithmes non polynomiaux est intéressante : en effet, le fait de faire des choix irrévocables sans connaître la totalité de l'instance débouche sur le fait qu'aucun algorithme on-line ne va pouvoir être optimal.

Le résultat principal concerne l'existence d'un algorithme (non polynomial) garantissant $\frac{1}{\sqrt{nt}}$ (où t est le nombre d'itérations nécessaires à la révélation de l'instance), et d'un algorithme polynomial garantissant $O\left(\frac{\log(n)}{n\sqrt{t}}\right)$. On obtient ainsi des garanties intéressantes si le nombre d'itérations est faible (en particulier s'il ne dépend pas de n). Détaillons l'obtention de ces résultats, puisque nous en réutiliserons l'idée par la suite. Le problème principal lors de la recherche d'un algorithme offrant une garantie de performance est la cécité de celui-ci ; ne connaissant pas la suite de l'instance, il se peut que, dès qu'il choisit un sommet, tous les sommets suivants soient reliés au sommet choisi et ainsi l'algorithme ne peut augmenter sa solution. Il semble donc que, pour pouvoir obtenir une garantie de performance intéressante, un algorithme ne doit prendre des sommets que s'il peut en prendre au moins un certain nombre. Considérons un algorithme pour le problème du stable classique (off-line), et considérons l'algorithme on-line qui, à chaque fois qu'un sous-ensemble de sommets S_i arrive :

- fait agir l'algorithme off-line sur S_i ;
- renvoie cette solution si sa taille est plus grande qu'un certain seuil, attend l'itération suivante sinon.

C'est en réglant de manière optimale le seuil, et en prenant pour algorithme off-line un algorithme exact (non polynomial) puis le meilleur algorithme polynomial connu que l'on obtient les deux résultats précédents.

Les auteurs montrent également qu'aucun algorithme (quelle que soit sa complexité) ne peut garantir mieux que $\frac{1}{\sqrt{n}-1}$, prouvant ainsi que l'algorithme proposé ne peut être sensiblement amélioré si t est constant (on ne peut faire mieux qu'une garantie en $O\left(\frac{1}{\sqrt{n}}\right)$).

L'objectif ici va être de définir et d'analyser, toujours en terme de recherche de garanties de performance, deux nouvelles approches du problème du stable on-line.

Dans la première, on va modifier le mode de révélation de l'instance. L'instance initiale va cette fois ci être un graphe complet, et ce sont les arêtes qui vont disparaître, jusqu'à l'obtention du graphe final. On aboutira pour ce modèle à des garanties similaires à celles trouvées pour le premier modèle.

Ensuite, on reprendra le modèle initial en le relâchant. Une première idée de relaxation, proposée dans [Halldorsson, 2002], consiste à permettre à l'algorithme de garder à chaque itération plusieurs solutions réalisables. Nous introduirons ici une possibilité de backtracking: l'algorithme pourra revenir sur certaines de ces décisions, cette nouvelle liberté étant compensée par l'instauration d'un système de pénalisation. Nous verrons ainsi comment on peut améliorer les garanties de performance précédentes

2 - MODELE DE DISPARITION DES ARETES

On va donc considérer ici le modèle suivant : le graphe initial est complet, et à chaque itération un ensemble d'arêtes disparaît. L'algorithme peut, à chaque itération, rajouter dans sa solution les sommets adjacents à (au moins) une arête supprimée. On considère bien entendu que quand l'algorithme choisit de mettre un sommet dans sa solution, ce choix est irrévocable.

Soit *FA* un algorithme pour le problème off-line.

Théorème 1 : *Si FA garantit* $\rho(n)$ *(décroissant en n), alors, en supposant que le nombre d'itérations est connu au départ, il existe un algorithme on-line OLTA garantissant* $\sqrt{\frac{\rho(n)}{nt}}$. *De plus, si FA est polynomial, OLTA l'est aussi.*

En particulier, en utilisant un algorithme *FA* exact, on obtient une garantie de $\frac{1}{\sqrt{nt}}$. On obtient également une garantie polynomiale en $O\left(\frac{\log(n)}{n\sqrt{t}}\right)$. Ces résultats sont particulièrement intéressants si le nombre d'itérations est faible (constant notamment).

Schéma de la preuve :

L'algorithme *OLTA* est une adaptation de celui présenté dans [Demange, 2000]. Le problème est en effet similaire ici : l'algorithme est dans une situation de cécité, la notion de seuil est encore pertinente. On va donc adopter une stratégie analogue : à chaque itération, on utilise un algorithme off-line et, si la solution fournie par celui-ci est plus petite que le seuil, on ne prend aucun sommet et on attend l'itération suivante. En réglant le seuil optimalement, on va alors pouvoir montrer que soit on aura une solution de taille conséquente, soit le graphe aura un nombre de stabilité faible, et aboutir ainsi au résultat.

Théorème 2 : *Même s'il n'y a que deux itérations, aucun algorithme ne peut garantir mieux que* $\frac{1}{\sqrt{n/2}-1}$.

Cette borne supérieure est de l'ordre de la garantie trouvée précédemment lorsque le nombre d'itérations est constant, ce qui indique que dans un tel cas on ne peut faire réellement mieux que l'algorithme à seuil *OLTA*.

Schéma de la preuve :

Il s'agit ici de "piéger" tout algorithme on-line *A*. Pour cela, on va révéler le graphe *G* (comportant *n* sommets $v_1, v_2, \dots, v_n$) en deux itérations. La première consiste à supprimer toutes les arêtes reliant deux sommets parmi $V=\{v_1, v_2, \dots, v_p\}$, où $p = \lfloor \sqrt{2n} \rfloor$, de manière à ce que *V* soit un stable.

La deuxième itération dépend des choix faits par l'algorithme *A* à l'issue de la première. Si *A* a choisi au moins un sommet (nécessairement dans *V*), on supprime les arêtes non adjacentes à un sommet de *V*. Dans ce cas, $|A(G)| \le p$ et $\alpha(G) \ge n-p$. Sinon, on supprime une seule arête, entre v_1 et v_n par exemple. Alors $|A(G)| \le 2$ et $\alpha(G) \ge p$. La valeur de *p* permet de montrer que, dans chacun des deux cas, $\frac{|A(G)|}{\alpha(G)} \le \frac{1}{\sqrt{n/2}-1}$.

Remarques :

- Si l'on s'intéresse aux garanties en fonction du degré Δ du graphe, on peut facilement obtenir, en définissant un algorithme (polynomial) de type glouton, une garantie de $\frac{1}{\Delta}$.
- Nous avons supposé ici que le nombre d'itérations est connu au départ par l'algorithme. Si tel n'est pas le cas, on peut alors montrer qu'aucun algorithme ne peut garantir mieux que $\frac{2}{n-1}$, ce qui est donc très faible.

3 - RELAXATION PAR UNE POSSIBLITE DE BACKTRACKING

Dans cette partie, nous reprenons donc le modèle initial où l'on part d'un graphe vide et où les sommets arrivent par paquets. Nous allons relâcher ce modèle en considérant que les choix faits par l'algorithme ne sont plus irrévocables. Un algorithme pourra ainsi choisir un sommet apparu à une itération précédente, ou sortir de la solution un sommet qu'il avait précédemment choisi.

Néanmoins, cette relaxation est trop forte puisqu'il suffit alors d'attendre la dernière itération et l'on est ramené au problème off-line. Ceci nous a conduit à l'idée d'instaurer des pénalités : si l'algorithme choisit un sommet révélé à une itération précédente, celui-ci ne vaudra plus une unité pleine mais aura une valeur entre 0 et 1, susceptible d'évoluer au cours du processus de révélation du graphe.

Plus précisément, nous allons définir un paramètre réel $k \in \left]1,\infty\right[$. La valeur d'un sommet apparu à l'itération i et choisi à l'itération $j \geq i$ sera $\frac{1}{k^{j-i}}$. Le but du problème est alors de construire un stable de valeur maximale, cette valeur étant bien entendu définie comme la somme des valeurs des sommets du stable.
On mesurera ici encore la qualité de la solution d'un algorithme on-line *OLA* sur une instance *G* en étudiant le rapport $\frac{val(OLA(G))}{\alpha(G)}$ (où *val(S)* désigne la valeur d'un stable *S*).

Maintenant, l'algorithme a donc face à lui à l'itération i un graphe valué de la manière suivante :

- Les sommets qui viennent d'arriver valent 1.
- Les sommets révélés à l'itération précédente valent $\frac{1}{k}$.
- …
- Les sommets révélés à la première itération valent $\frac{1}{k^{i-1}}$.

Il peut choisir n'importe quels sommets de ce graphe. Il semble alors naturel d'utiliser maintenant un algorithme off-line pour le problème du stable pondéré. Soit *WFA* un tel algorithme :

Théorème 3 : *Si WFA garantit ψ(n) (décroissant en n), alors il existe un algorithme on-line WOLA garantissant* $\frac{\psi(n)}{t(1-1/k)-1/k}$. *De plus, WOLA est polynomial si WFA l'est.*

On obtient donc une nette amélioration des garanties de performance par rapport au modèle non relâché. On obtient en particulier un algorithme (non polynomial) en $\frac{1}{t}$, donc à rapport constant si t est constant.

Schéma de la preuve :

L'algorithme *WOLA* consiste simplement à faire agir à chaque itération *WFA* sur le graphe valué dans son ensemble. Si la solution est meilleure que la solution courante, alors on prend cette nouvelle solution comme solution courante (ce qui est permis étant donné que les choix ne sont plus irrévocables).

L'obtention de la garantie de performance est un peu longue, mais elle consiste à :

- majorer la valeur d'un stable de poids maximal dans les graphes intermédiaires en fonction de la solution fournie par *WOLA* (ce qui est immédiat) ;
- majorer le nombre de stabilité du graphe en fonction des stables de poids maximal des graphes intermédiaires.

On peut maintenant légitimement se demander si l'on ne pourrait pas améliorer sensiblement cette garantie. La réponse est négative (du moins si le nombre de sommets est "assez grand"), puisque l'on a le théorème suivant :

Théorème 4 : *Si* $n \geq \frac{t(t+1)}{2}$, *aucun algorithme ne peut garantir strictement mieux que* $\frac{1}{t(1-1/k)}$.

La borne supérieure est donc quasiment égale à la garantie trouvée précédemment.

Schéma de la preuve :

Il va ici encore falloir piéger tout algorithme on-line *OLA*. Nous allons construire notre instance de la manière suivante : le premier sous-ensemble de sommets sera une clique de taille *t*, le deuxième une clique de taille $t-1$, ..., l'avant dernier une clique de taille 2, et le dernier une clique formée des *n-t(t-1)/2* sommets restants (d'où l'hypothèse sur *n* dans l'énoncé du théorème). Pour les arêtes, on relie à chaque itération tous les sommets de la nouvelle clique à tous les sommets de la solution courante. On peut alors montrer que dans ce graphe :

- il existe un stable de taille *t* ;
- $val(OLA(G)) \leq 1 + \frac{1}{k} + ... + \frac{1}{k^{t-1}} \leq \frac{1}{1-1/k}$

Ces deux propriétés permettent d'obtenir le résultat annoncé.

Remarque :

On pourrait envisager une relaxation plus limitée en considérant que le choix de mettre un sommet dans la solution demeure irrévocable (mais en se permettant de prendre des sommets que l'on avait précédemment refusés). Néanmoins, dans un tel modèle, le problème de cécité de l'algorithme reste entier et l'on ne parvient à améliorer que peu sensiblement la garantie obtenue dans le modèle non relâché.

4 - CONCLUSION

L'étude de problèmes on-line montre comment ces problèmes relativement nouveaux constituent un cadre naturel pour un prolongement de la théorie de l'approximation classique. On y recherche ainsi des résultats positifs (des garanties de performance) et négatifs (des bornes sur les garanties possibles), le but étant bien entendu de minimiser l'écart entre la meilleure garantie trouvée et la "meilleure" borne. On y retrouve également la notion clé de réduction en approximation permettant d'obtenir une garantie de performance pour un problème en fonction de celle d'un autre (ici entre les versions on-line et off-line de ce problème).

Il pourrait être intéressant d'élargir cette notion de réduction en approximation en réduisant un problème on-line à un autre problème off-line, ou même à un autre problème on-line.

Une deuxième voie pourrait être d'appliquer les modèles vus ici sur le stable (la relaxation notamment) à d'autres problèmes d'optimisation on-line, comme la coloration.

Enfin, il serait particulièrement intéressant et utile d'essayer d'unifier les différentes approches dans l'étude des problèmes on-line, les différents modèles, de proposer par exemple une sorte de classification permettant de définir clairement des problématiques, de manière à éviter un éparpillement dû au grand nombre de modélisations possibles.

BIBLIOGRAPHIE

[Ausiello, 2001] Ausiello G., Feuerstein E., Leonardi S., Stougie L., Talamo M.: “Algorithms for the on-line travelling salesman problem”. Algorithmica, Vol 29, N°4, p. 560-581 (2001).

[Boppana, 1992] Boppana R., Halldorsson M. M.: “Approximating maximum independent sets by excluding subgraphs”. BIT, Vol 32, N°2, p. 180-196 (1992).

[Demange, 2000] Demange M., Paradon X., Paschos V. Th. : “On-line maximum-order induced hereditary subgraph problem”. SOFSEM, p. 327-335 (2000).

[Halldorsson, 2000a] Halldorsson M. M. : “Approximations of weighted independent set and hereditary subset problems”. Journal of graph algorithms and applications, Vol 4, N°1, p. 1-16 (2000).

[Halldorsson, 2000b] Halldorsson M. M. : “Online coloring known graphs”. Electronic Journal of Combinatorics, Vol 7, (2000).

[Halldorsson, 1994] Halldorsson M. M., Szegedy M. : “Lower bounds for on-line graph coloring”. Theoretical Computer Science, Vol 130, N°1, p. 163-174 (1994).

[Halldorsson, 2002] Halldorsson M. M., Iwama K., Miyazaki S., Taketomi S. : “Online independent sets”. Theoretical Computer Science, Vol 289, N°2, p. 953-962 (2002).

[Paradon, 2000] Paradon X. : “Algorithmique on-line”. Thèse de doctorat, Université Paris Dauphine (2000).

[Paschos, 2001] Paschos V. Th. : “on-line independent set by coloring vertices”. Proceedings of the 16th symposium on computer and information sciences (2001).