

Améliorer la complexité de la résolution de problèmes difficiles : le cas de la clique dominante

Nicolas Bourgeois¹, Federico Della Croce², Bruno Escoffier¹, Vangelis Paschos¹

¹ Lamsade, Université Paris Dauphine et CNRS FRE 32-34, Place du Marechal de Lattre de Tassigny, Paris, France

`prenom.nom@dauphine.fr`

² D.A.I., Politecnico di Torino, Torino, Italie

`federico.dellacroce@polito.it`

Mots-Clés : *Complexité au pire des cas, problèmes NP-difficiles, clique dominante.*

Pour faire face à l'existence de problèmes NP-complets, il existe principalement deux approches classiques, à savoir la résolution exacte (au prix d'une complexité élevée) et la recherche d'heuristiques, autrement dit d'algorithmes plus rapides susceptibles de produire des solutions "proches" de l'optimal. Ce sont là deux champs de recherche très actifs tant en optimisation combinatoire qu'en informatique théorique. En ce qui concerne la résolution exacte, de nombreuses méthodes ont été développées depuis les débuts de l'optimisation combinatoire pour produire des algorithmes aussi efficaces que possibles. Cela dit, ce n'est que récemment - depuis une petite dizaine d'années - que les chercheurs ont essayé de proposer des garanties de performance en terme de temps de calcul (complexité au pire des cas) sur ces méthodes exactes. Ainsi, de nombreux et récents travaux se sont consacrés au développement d'algorithmes résolvant un problème NP-complet mais ayant une complexité au pire des cas la plus faible possible. En considérant par exemple le très classique problème du stable maximum (trouver dans un graphe un ensemble aussi grand que possible de sommets deux à deux non adjacents), il est facile de voir qu'un algorithme examinant successivement tous les sous-ensembles de sommets et conservant chaque fois le plus grand produira la solution optimale en un temps $O^*(2^n)$, où n est le nombre de sommets du graphe - la notation $f = O^*(g)$ signifiant que l'on néglige un éventuel facteur polynomial. La question qui se pose alors est de savoir si l'on peut concevoir un algorithme produisant un stable de cardinal maximal en un temps n'excédant pas $O^*(\gamma^n)$, avec $\gamma < 2$. La réponse est oui, et plusieurs articles récents se sont attachés à abaisser toujours davantage la valeur de cette constante (voir [4, 5] par exemple).

Nous nous intéressons ici au problème de la clique dominante dans un graphe. Soit $G = (V, E)$ un graphe. Une clique de G est un ensemble de sommets $V' \subset V$ deux-à-deux adjacents (induisant donc un sous-graphe complet). Un ensemble dominant est un sous-ensemble de sommets $V' \subset V$ tel que tout sommet $v \in V$ soit dans V' soit est voisin d'un sommet de V' . Le problème de savoir s'il existe une clique dominante dans un graphe est NP-complet [2]. Comme pour le problème du stable maximum, un examen exhaustif des sous-ensembles de sommets permet de résoudre le problème en un temps $O^*(2^n)$. Ce résultat peut être amélioré grâce à une propriété sur le nombre de cliques maximales pour l'inclusion dans un graphe [3], et a été nettement amélioré récemment dans [1] où un algorithme ayant une complexité de $O^*(1,34^n)$ a été proposé. Dans ce travail, nous poursuivons l'amélioration de la complexité de résolution de ce problème en proposant un algorithme de complexité $O^*(1,28^n)$. Nous étudions également deux variantes de ce problème consistant à trouver - quand il en existe - une clique dominante de taille respectivement maximale et minimale. Pour ces deux problèmes, évidemment au moins aussi longs à résoudre que le problème de l'existence, nous

proposons des algorithmes ayant respectivement une complexité de $O^*(1.32^n)$ et $O^*(1.33^n)$. Nous montrons également que tous ces résultats peuvent être améliorés lorsque des hypothèses sont faites sur la densité du graphe, et proposons pour les deux versions optimisation des algorithmes approchés ayant une complexité plus faible que les algorithmes exacts correspondants.

Références

- [1] Kratsch, D., Liedloff, M. : An exact algorithm for the minimum dominating clique problem. *Theoret. Comput. Sci.* **385** (2007) 226–240
- [2] Garey, M.R., Johnson, D.S. : *Computers and intractability. A guide to the theory of NP-completeness*. W. H. Freeman, San Francisco (1979)
- [3] Johnson, D.S., Yannakakis, M., Papadimitriou, C.H. : On generating all maximal independent sets. *Inform. Process. Lett.* **27** (1988) 119–123
- [4] J. M. Robson. Finding a maximum independent set in time $O(2^{n/4})$. Technical Report 1251-01, LaBRI, Université de Bordeaux I, 2001.
- [5] G. J. Woeginger. Exact algorithms for NP-hard problems : a survey. In M. Juenger, G. Reinelt, and G. Rinaldi, editors, *Combinatorial Optimization - Eureka! You shrink!*, volume 2570 of *Lecture Notes in Computer Science*, pages 185–207. Springer-Verlag, 2003.