Planification d'horaires de travail et colorations de graphes

Frédéric GARDI[1,2]

1. *Laboratoire d'Informatique Fondamentale, Parc Scientifique et Technologique de Luminy, Case 901 - 163, avenue de Luminy, 13288 Marseille Cedex 9, France*

2. Prologia - Groupe Air Liquide*, Parc Scientifique et Technologique de Luminy, Case 919 - 163, avenue de Luminy, 13288 Marseille Cedex 9, France*

Frederic.Gardi@lif.univ-mrs.fr

Mots-clefs : planification d'horaires de travail, graphes d'intervalles et d'arcs circulaires, colorations.

Étudiée depuis plusieurs décennies, la *planification d'horaires de travail* est un problème classique de recherche opérationnelle [1]. Il est plus que jamais d'actualité du fait de la réduction du temps de travail en France. De manière schématique, le problème consiste en l'affectation d'un ensemble de tâches fixes à un ensemble d'employés sous forme de vacations, une vacation se définition comme une succession de tâches *deux-à-deux disjointes* dans le temps [2]. L'*objectif* de cette affectation est de minimiser le nombre de vacations nécessaire à l'exécution de toutes les tâches par souci de *productivité*. La difficulté du problème est que la construction des vacations doit respecter certaines contraintes réglementaires liées au code du travail. Les réglementations étant aussi nombreuses que les métiers auxquelles elles s'appliquent, les modèles de planification peuvent ainsi se décliner à l'infini. Aussi nous n'aborderons pas ici la résolution d'un cas pratique et singulier de planification, mais notre étude portera plutôt sur le problème fondamental suivant:

Soit un ensemble $\{T_i\}_{i=1,\dots,n}$ de tâches possédant chacune une date de début d_i et une date de fin f_i. Soit k un entier naturel. Partitionner ces n tâches en le minimum de vacations tel que les tâches de chacune ne se recouvrent pas entre elles et que le nombre de tâches qu'elles contiennent soit limité à k.

Les tâches étant de simples intervalles sur la droite du temps, ce problème peut être reformulé du point de vue de la théorie des graphes comme le problème de la *coloration minimum* d'un graphe d'intervalles tel que chaque couleur marque au plus k sommets. Un graphe non-orienté $G = (V, E)$ est un *graphe d'intervalles* si à chaque sommet $v \in V$ peut être associé un intervalle $I_v = [d_v, f_v]$ tel que pour chaque paire de sommets $u, v \in V$, u et v sont adjacents ssi I_u et I_v s'intersectent. La famille $\{I_v\}_{v\in V}$ est appelé *représentation par intervalles* de G. Nous considérerons qu'une telle représentation (triée par extrémités) est fournie en entrée de nos algorithmes, celle-ci pouvant être calculé en $O(n + m)$, avec $n = |V|$ et $m = |E|$ [3, 4] . Un graphe est dit *d'intervalles propres* s'il possède une représentation par intervalles où aucun intervalle n'en inclut strictement un autre. Dans le cas de tâches *cycliques*, le même problème peut se définir sur les *graphes d'arcs circulaires*, une extension naturelle des graphes d'intervalles (les intervalles devenant des arcs sur un cercle). Il est à noter que ces classes de graphes interviennent dans de nombreuses applications pratiques; en effet, elles servent à modéliser des structures apparaissant dans des domaines aussi divers que la génétique, l'ordonnancement, la psychologie, la sociologie ou encore l'archæologie. Pour plus de détails sur ces graphes et leurs applications, le lecteur pourra se référer à [5, 3, 6].

Le problème de coloration classique (sans contrainte, $k \geq n$) a été étudié il y a déjà plusieurs décennies. Pour les graphes d'intervalles, Gupta *et al* [7] donne un algorithme en $O(n \log n)$, tandis que Garey *et al* montre que le problème est $\mathcal{NP}$-difficile pour les graphes d'arcs [8]. Toutefois, pour les graphes d'arcs propres, Shih et Hsu [9] donne un algorithme en $O(n^{1.5})$. Lorsque $k \leq 2$, l'utilisation d'un algorithme de couplage maximum [10] sur le graphe complément permet de résoudre en $O(n^{2,5})$ le problème (même pour des graphes quelconques). Plus récemment, Bodlaender et Jansen [11] ont montré que pour $k \geq 4$ fixé, le problème devient $\mathcal{NP}$-difficile pour les graphes d'intervalles; Andrews *et al* [12] décrivent de leur côté un algorithme récursif en $O(n \log n)$ pour ces mêmes graphes lorsque $k = 2$. Dans le tableau ci-dessous (Fig. 1), nous récapitulons tous ces résultats et nous les complétons par ceux de nos recherches en cours. Tout d'abord, nous donnons un algorithme simple et incrémental en $O(n)$ pour les graphes d'intervalles lorsque $k = 2$, ainsi qu'un algorithme en $O(n)$ pour les graphes d'intervalles propres quelque soit k [13]. Ensuite, nous proposons un algorithme en $O(n^2)$ pour les graphes d'arcs propres avec k quelconque, ainsi qu'un algorithme en $O(n)$ pour ces mêmes graphes et $k = 2$ [14]. Pour conclure, nous présentons les quelques questions restant ouvertes sur le sujet.

	Intervalles propres	Intervalles	Arcs propres	Arcs
$k = 2$	$O(n)$ [12, 13]	$O(n)$ [12, 13]	$O(n)$ [14]	$O(n^{2.5})$ [10]
$k = 3$	$O(n)$ [13]	*ouvert*	$O(n^2)$ [14]	*ouvert*
$k \geq 4$	$O(n)$ [13]	NPc [11]	$O(n^2)$ [14]	NPc [11]

Fig. 1. Complexité du problème selon la classe du graphe et la valeur de k.

References

[1] A. Partouche (1998). *Planification d'horaires de travail.* Thèse de Doctorat de 3ème cycle, Université Paris-Dauphine.

[2] Bamboo–Planification by Prologia - Groupe Air Liquide. http://prologianet.univ-mrs.fr/bamboo/bamboo_planification.html

[3] M.C. Golumbic (1980). *Algorithmic Graph Theory and Perfect Graphs.* Computer Science and Applied Mathematics, Academic Press, New-York, NY.

[4] R.M. McConnell (2001). Linear-time recognition of circular-arc graphs. In *Proceedings of FOCS 2001*, pages 386–394.

[5] F.S. Roberts (1978). *Graph Theory and its Application to the Problems of Society.* SIAM Publications, Philadelphia, PA.

[6] C. Berge (1984). *Graphes.* Gauthier-Villars, Paris.

[7] U.I. Gupta, D.T. Lee et J.Y.-T. Leung (1982). Efficient algorithms for interval graphs and circular-arc graphs. *Networks* 12, 459–467.

[8] M. Garey, D. Johnson, G. Miller et C. Papadimitriou (1982). The complexity of coloring circular-arc graphs and chords. *SIAM Journal on Discrete Mathematics* 1(2), 216–227.

[9] W.-K. Shih et W.-L. Hsu (1989). An $O(n^{1.5})$ algorithm to color proper circular-arcs. *Discrete Applied Mathematics* 25(3), 321–323.

[10] S. Micali et V.V. Vazirani (1980). An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs. In *Proceedings of FOCS 1980*, pages 17–27.

[11] H.L. Bodlaender et K. Jansen (1995). Restrictions on graph partition problems. Part I. *Theoretical Computer Science* 148, 93–109.

[12] M.G. Andrews, M.J. Atallah, D.Z. Chen et D.T. Lee (2000). Parallel algorithms for maximum matching in complements of interval graphs and related problems. *Algorithmica* 26, 263–289.

[13] F. Gardi (2003). Efficient algorithms for disjoint matchings among intervals and related problems. In *Proceedings of DMTCS 2003*, *LNCS* 2731, pages 268–280.

[14] F. Gardi. Mutual exclusion scheduling with interval graphs or related classes. Part I. (en préparation).