Planification d'horaires de travail et théorie des 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, *Parc Scientifique et Technologique de Luminy, Case 919 - 163, avenue de Luminy, 13288 Marseille Cedex 9, France*

frederic.gardi@lidil.univ-mrs.fr

Mots-clefs : planification d'horaires de travail, optimisation multicritères, graphes d'intervalles, colorations.

Etudiée depuis plusieurs décennies, la *planification d'horaires de travail* est un problème classique de recherche opérationnelle [1]. Il est d'autant plus d'actualité du fait de la réduction du temps de travail en France. De manière schématique, ce problème consiste en l'affectation d'un ensemble de tâches à un ensemble d'employés sous la forme de vacations. Les *critères d'optimisation* de cette affectation sont alors : minimiser le nombre de vacations (*productivité*), équilibrer le planning (*social*) et prévenir au mieux les modifications de planning (*robustesse*). La difficulté du problème est que la construction des vacations doit respecter un certain nombre de contraintes liées aux règlementations du travail. Ces règlementations étant aussi nombreuses que les métiers auxquelles elles s'appliquent, les modèles de planification peuvent se décliner à l'infini. Aussi nous n'aborderons pas dans ce papier la résolution d'un cas pratique et singulier de planification. Notre étude porte au contraire sur le problème fondamental suivant :

Soit un ensemble $\{T_i\}_{i=1,\ldots,n}$ 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 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 pouvant être vues comme de simples intervalles sur une droite, notre problème peut être reformulé en les termes de la théorie des graphes comme le problème de la *coloration d'un graphe d'intervalles* tel que chaque couleur marque au plus k sommets (problème baptisé CGI_k). Un ensemble de sommets de même couleur étant appelé *stable*, celui-ci peut encore être défini comme un problème de partition d'un graphe d'intervalles par des stables de taille au plus k. 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 une *représentation par intervalles* de G. Un graphe est appelé *graphe d'intervalles propres* s'il existe une représentation par intervalles de celui-ci où aucun intervalle n'est strictement inclu dans un autre. Les graphes d'intervalles interviennent dans de nombreuses applications pratiques ; ils modélisent des structures apparaissant dans des domaines aussi diverses que la génétique, l'ordonnancement, la psychologie, la sociologie ou encore l'archæologie. Enfin, beaucoup de problèmes $\mathcal{NP}$-complets pour des graphes arbitraires (coloration minimum, clique maximum, etc [2]) se résolvent en temps polynomial pour les graphes d'intervalles [3]. Pour plus de détails sur ces graphes et les notions de théorie des graphes employées dans ce papier, le lecteur pourra se référer à [4, 5].

Les critères d'optimisation définis précédemment peuvent alors être modélisés comme suit : (C_1) minimiser le nombre de couleurs, (C_2) équilibrer le nombre d'intervalles de chaque couleur, (C_3) maximiser l'écart minimum entre deux intervalles consécutifs d'une même couleur. Ces critères, comme bien souvent en optimisation multicritères, sont contradictoires. Nous noterons que le critère C_3 permet de prévenir les chevauchements de tâches lorsque celles-ci seront retardées ou avancées. Partitionner un graphe d'intervalles en un minimum de stables de taille au plus k est un problème $\mathcal{NP}$-complet même si la donnée k est une constante égale à quatre [6]. De là, optimiser le problème CGI_k selon les seuls critères C_1 ou C_2 est aussi un problème $\mathcal{NP}$-complet. A notre connaissance, la complexité de ce problème pour une constante $k = 3$ reste une question ouverte. Pour $k = 2$, des techniques de couplage permettent de le résoudre en temps linéaire [7, 8]. En définitive, le problème de la planification d'horaires, même modélisé simplement et sauf cas particulier, reste un problème *difficile*.

Dans une première partie, nous montrons qu'un algorithme glouton permet de résoudre de manière optimale et en temps linéaire le problème CGI_k pour les critères C_1, C_2, C_3 si le graphe donné en entrée est un graphe d'intervalles propres. En particulier, nous établissons que la partition optimale d'un graphe d'intervalles propres G par des stables de taille au plus k a pour cardinalité la borne inférieure $\max(\omega(G), \lceil \frac{n}{k} \rceil)$ où $\omega(G)$ est la taille d'une clique maximum de G et n son nombre de sommets. Une idée serait donc de traiter le problème CGI_k en partitionnant le graphe d'intervalles de départ en plusieurs sous-graphes d'intervalles propres, une application de l'algorithme glouton sur chacun de ces sous-graphes permettant de résoudre *localement* le problème. Bien évidemment, la qualité d'une telle solution dépend fortement de la manière dont le graphe d'intervalles est partitionné en sous-graphes d'intervalles propres. Dans une seconde partie, nous montrons alors qu'il est possible de déterminer en temps linéaire une partition en sous-graphes d'intervalles propres garantissant une $O(\log n)$-approximation pour le critère C_1. Pour cela, nous établissons tout d'abord que pour tout entier $t \geq 3$, un graphe d'intervalles sans $K_{1,t}$ peut être partitionné en $\lfloor \frac{t}{2} \rfloor$ sous-graphes d'intervalles propres. Ce lemme nous permet de prouver que tout graphe d'intervalles est partitionnable en $2\lceil \log_3 n \rceil$ sous-graphes d'intervalles propres. De plus, la preuve du résultat fournit un algorithme linéaire pour calculer cette partition. Enfin, nous exhibons une famille de graphes d'intervalles pour laquelle cette borne est quasiment atteinte. Pour conclure, nous montrons qu'en situation réelle (problèmes de planification de personnel d'aérogares traités par l'entreprise PROLOGIA–Groupe Air Liquide), cette même borne n'est plus logarithmique mais constante, puis discutons les résultats obtenus.

Références

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

[2] M.R. Garey et D.S. Johnson (1979). *Computer and Intractability : A Guide to $\mathcal{NP}$-Completeness.* W.H. Freeman.

[3] 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.

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

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

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

[7] 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.

[8] F. Gardi (2002). Efficient algorithms for disjoint matchings among intervals and related problems. *En préparation*, version étendue du Rapport de Recherche 07-2002 du LIF.