Un algorithme exponentiel basé sur Inclusion-Exclusion pour la résolution d'un problème d'ordonnancement de type flowshop

Olivier Ploton[1], Vincent T'kindt[1]

Université de Tours, Laboratoire d'Informatique Fondamentale et Appliquée (LIFAT, EA 6300), ERL CNRS 7002 ROOT, Tours, France

{olivier.ploton,vincent.tkindt}@univ-tours.fr

Mots-clés : *ordonnancement, algorithmique exponentielle, inclusion-exclusion.*

1 Introduction

Dans ce travail nous nous intéressons à un problème d'ordonnancement de n travaux dans un environnement de type flowshop à m machines. Chaque travail i est défini par une durée opératoire p_{ij} lorsqu'il est traité par la machine j. Tous les travaux sont supposés passer dans le même ordre sur chaque machine et l'objectif est de minimiser la date de fin globale, définie par $C_{max} = \max_{1 \leq i \leq n}(C_{i,m})$ avec $C_{i,j}$ la date de fin du travail i sur la machine j dans l'ordonnancement. Ce problème est noté $F|prmu|C_{max}$ ([4]) et est $\mathcal{NP}$-difficile au sens fort.

Nous nous intéressons à la résolution exacte de ce problème et notamment à l'établissement de bornes sur les complexités temporelles et spatiales au pire cas. Pour le $F3|prmu|C_{max}$, Shang et al. ([9]) ont proposé un algorithme de programmation dynamique fonctionnant en $O^*(3^n)$ en temps et en espace. Cette complexité au pire cas peut également être écrite sous la forme $O^*(2^n p_{max})$ avec $p_{max} = \max_{i,j}(p_{ij})$. Nous montrons, grâce à la technique d'inclusion-exclusion, que le $F3|prmu|C_{max}$ peut être résolu en espace pseudo-polynomial sans dégrader la complexité temporelle au pire cas. Nous montrons également que ce résultat peut être généralisé dans le cas où m est quelconque.

La formule d'inclusion-exclusion exprime le cardinal d'une union de N ensembles en fonction des cardinaux des intersections de ces ensembles. L'intérêt de cette formule est qu'elle permet de décider si un problème admet une solution sans jamais en expliciter. Cette technique, décrite dans [3], a été appliquée à de nombreuses reprises à des problèmes NP-difficiles comme le calcul du permanent [8], les chemins hamiltoniens [1, 5], le recouvrement ou partitionnement d'ensembles [2, 6]. Cette technique a été peu utilisée en ordonnancement ([5, 7]).

2 Problèmes de permutation et inclusion-exclusion

En ordonnancement on est souvent confronté à des problèmes de permutation : étant donné un ensemble de n travaux, trouver dans quel ordre les agencer pour minimiser un objectif. L'algorithme brutal consistant à énumérer tous les ordres possibles est en $O^*(n!)$.

Pour présenter la technique d'inclusion-exclusion, supposons qu'une solution s'exprime comme un couple (π, S) vérifiant une contrainte de validité $\mathcal{V}$, où π est une permutation de $\{1 \dots n\}$ décrivant l'ordre des travaux, et S la liste des dates de début des travaux. Si l'on autorise des travaux à apparaître plusieurs fois dans un ordonnancement (π, S), autrement dit si π est une application et pas forcément une permutation, alors les solutions sont les couples valides (π, S) pour lesquels π est une permutation ($\text{Im } \pi = \{1 \dots n\}$), et l'inclusion-exclusion donne :

$$\underbrace{\text{card}\left\{(\pi, S) \,\middle|\, \begin{array}{l} \mathcal{V}(\pi, S) \\ \text{Im } \pi = \{1 \dots n\} \end{array}\right\}}_{\text{nombre de solutions}} = \sum_{X \subset \{1 \dots n\}} (-1)^{n-|X|} \underbrace{\text{card}\left\{(\pi, S) \,\middle|\, \begin{array}{l} \mathcal{V}(\pi, S) \\ \text{Im } \pi \subset X \end{array}\right\}}_{N_X \text{ à calculer}} \qquad (1)$$

Souvent, chaque terme N_X peut se calculer à X fixé par programmation dynamique en temps et espace polynomial. Au final, le calcul complet s'effectue en temps $O^*(2^n)$ et espace polynomial.

3 Application au problème du flowshop

Pour une valeur donnée du $C_{\max}$, notée ϵ, un ordonnancement (π, S) respecte la contrainte de validité $\mathcal{V}$ si et seulement si il respecte les contraintes de gamme, les contraintes disjonctives et si $C_{\max}(\pi, S) \leq \epsilon$. La contrainte $\text{Im } \pi = \{1 \dots n\}$ dans la formulation (1) implique que dans (π, S) chaque travail apparaît une et une seule fois. Nous proposons un algorithme exact dont la complexité temporelle au pire cas, à m fixé, est en $O^*(2^n ||I||^{O(1)})$ où $||I||$ est une mesure de la valeur des éléments de l'instance (par exemple $p_{max} = \max_{i,j}(p_{ij})$). Sa complexité spatiale est en $O(||I||^{O(1)})$.

La structure de l'algorithme s'articule en trois étapes :

— Grâce à la technique d'inclusion-exclusion on peut compter le nombre de solutions et ainsi résoudre le problème de décision lorsque ϵ est fixé.

— Le problème d'optimisation consiste à trouver $C_{\max}^{\text{opt}}$, la plus petite valeur ϵ pour laquelle il existe une solution. Celle-ci est comprise entre 0 et la somme des p_{ij}, et on la détermine par dichotomie en $O(\ln ||I||)$ étapes.

— Pour construire une solution explicite, on considère la permutation $\pi = (\pi_1 \dots \pi_n)$ et $\epsilon = C_{\max}^{\text{opt}}$. On essaye les n valeurs possibles de π_n, ce qui ramène à n problèmes de dénombrement à $(n-1)$ travaux. Une fois un π_n possible déterminé, on procède récursivement pour $(\pi_1 \dots \pi_{n-1})$.

4 Conclusion

L'algorithme présenté montre qu'il est possible de résoudre, à m fixé, le problème $F|prmu|C_{\max}$ avec une garantie théorique de temps $O^*(2^n ||I||^{O(1)})$ et espace polynomial. Cela constitue une première contribution de la formulation inclusion-exclusion par rapport au résultat dans [9].

Références

[1] E.T. Bax. Inclusion and exclusion algorithm for the Hamiltonian path problem *Information Processing Letters*, 47(4) : 203–207, 1993.

[2] A. Björklund, T. Husfeldt. Inclusion-exclusion algorithms for counting set partitions *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006)*, 575–582, 2006.

[3] F.V. Fomin, D. Kratsch. Exact exponential algorithms. EATCS Springer, 2010

[4] R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan. Optimization and Approximation in Deterministic Sequencing and Scheduling : a Survey. *Proceedings of the ARI on Discrete Optimization and Systems Applications*, Elsevier, (5) : 287–326, 1979

[5] R.M. Karp. Dynamic Processing meets the principle of inclusion and exclusion. *Operational Research Letters*, 1(2) : 49–51, 1982.

[6] M. Koivisto. An $O(2^n)$ algorithm for graph coloring and other partitioning problems via inclusion-exclusion *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006)*, 583–590, 2006.

[7] J. Nederlof. Inclusion-exclusion for hard problems. Master Thesis. 2008

[8] H.J Ryser. Combinatorial mathematics. The Carus Mathematical Monographs, No 14. The Mathematical Association of America, 1963.

[9] L. Shang, C. Lenté, M. Liedloff, V. T'kindt. Exact exponential algorithms for 3-machine flowshop scheduling problems *Journal of Scheduling*, 21 : 227-233, 2018