EVRIPIDIS BAMPIS

The complexity of short schedules for uet bipartite graphs

<http://www.numdam.org/item?id=RO_1999__33_3_367_0>

THE COMPLEXITY OF SHORT SCHEDULES
FOR UET BIPARTITE GRAPHS (*)

by Evripidis Bampis ([1])

Communicated by Philippe Chrétienne

Abstract. – *We show that the problem of deciding if there is a schedule of length three for the multiprocessor scheduling problem on identical machines and unit execution time tasks in NP-complete even for bipartite graphs, i.e. for precedence graphs of depth one. This complexity result extends a classical result of Lenstra and Rinnoy Kan [5].*

Keywords: Schedule, multiprocessor, bipartite graph, complexity.

Résumé. – *Nous démontrons que le problème de décider s'il existe un ordonnancement de longueur trois pour le problème d'ordonnancement multiprocesseur, avec des machines identiques et des tâches de durée unitaire, est NP-complet même pour les graphes de précédence bipartis, i.e. les graphes de précédence de profondeur un. Ce résultat de complexité constitue une extension d'un résultat classique de Lenstra et Rinnoy Kan [5].*

Mots clés : Ordonnancement, multiprocesseur, graphe bipartit, complexité.

1. INTRODUCTION

This note deals with the difficulty of obtaining near-optimal solutions for the multiprocessor scheduling problem subject to precedence delays: Given n unit execution time tasks whose precedence constrains form a directed acyclic graph and m identical machines the objective is to find a schedule minimizing the makespan, *i.e.* the time at which the last task of the precedence graph completes its execution. Using the three field notation [3] this problem can be denoted as:

$$P|\text{prec}, p_j = 1|C_{\max}.$$

This problem has been extensively studied in the literature. Graham [1] proved that any list scheduling strategy is a 2-approximation algorithm. Ullman [8] proved that this problem is NP-hard. Furthermore, Lenstra and Rinnoy Kan [5] showed that the problem of deciding if there is a feasible schedule of length three is NP-complete, and thus there is no hope of finding an approximation algorithm with performance guarantee less than $\frac{4}{3}$, unless $P = NP$. The proof is based on a reduction of the CLIQUE problem and was presented in some of the most classical books of Computer Science (*see* for instance [6]). We extend this classical result by showing that it is true even if the precedence graph is a bipartite graph, *i.e.* a precedence graph with depth one.

2. THE NP-COMPLETENESS RESULT

In this section we prove the following theorem:

THEOREM: *The problem* $P|\text{prec} = \text{bipartite}, p_j = 1|C_{\max} \leq 3$ *is* NP-*complete*.

Proof: The proof is based on the balanced bipartite independent set problem (BBIS) whose NP-completeness is clearly implied by the well known NP-complete problem balanced bipartite complete graph (BBCG) [2].

Instance of BBIS: An undirected balanced bipartite graph $B = (X \cup Y, E)$, with $|X| = |Y| = n$, and an integer k.

Question: Is there in B, an independent set with k vertices in X and k vertices in Y?

If such an independent set exists, we call it *balanced independent set* of order k.

The problem remains NP-complete even if $k = \frac{n}{2}$ [9]. In the following we consider BBIS as the source problem.

Given any instance of BBIS, *i.e.* a balanced bipartite graph $B = (X \cup Y, E)$, with $|X| = |Y| = n$, we construct an instance of our scheduling problem in the following way:

There are $m = n$ identical processors to execute $3\,n$ unit execution time tasks. The precedence graph G that we consider contains:

1. $B = (X \cup Y, E)$, with $|X| = |Y| = n$, and where the edges between X and Y are replaced by the corresponding arcs directed from the vertices of X towards the vertices of Y.

2. A complete bipartite graph $A = (W \cup Z, E')$, with $|W| = |Z| = \frac{n}{2}$ and where all the vertices of W precede all the vertices of Z.

3. All the vertices of W precede all the vertices of Y and all the vertices of X precede all the vertices of Z.

Notice that G is bipartite, and that the corresponding instance of $P|\text{prec} = \text{bipartite}, p_j = 1|C_{\max} \leq 3$ may be computed in polynomial time.

We have now to show that B contains a balanced independent set with $\frac{n}{2}$ vertices in X and $\frac{n}{2}$ vertices in Y *if and only if* there is a feasible schedule of G with length three.

Let us first consider that B contains such a balanced independent set, call it $S = X_1 \cup Y_1$ with $X_1 \subset X$, $Y_1 \subset Y$, and $|X_1| = |Y_1| = \frac{n}{2}$. Then there is a schedule of G with length three:

$t = 1$: execute all the tasks of $X - X_1$ and all the tasks of W.

$t = 2$: execute all the tasks of X_1 and all the tasks of Y_1.

$t = 3$: execute all the tasks of $Y - Y_1$ and all the tasks of Z.

It is easy to verify that this schedule is feasible since all the precedence constraints are satisfied. The feasibility of the schedule is guaranteed from the existence of the a balanced independent set (X_1, Y_1).

Conversely, let us now assume that there is a feasible schedule of G of length three. We have to show that any schedule of this length necessitates the existence of a balanced independent set $X_1' \cup Y_1'$ with $X_1' \subset X$, $Y_1' \subset Y$, and $|X_1'| = |Y_1'| = \frac{n}{2}$ in B.

The key point of the proof is that all the tasks of W (resp. Z) have to be executed during the first (resp. third) time unit. This is clear since every task of W (resp. Z) precedes (resp. is preceded by) exactly $\frac{3n}{2}$ tasks and there is only n machines. Thus, only $\frac{n}{2}$ tasks of X can be executed at the first time unit. Notice that no task of Y can be executed in the first time unit because of the constraints coming from W. Similarly, all the task of X have to finish no later than the second time unit since they precede all the tasks of Z. Given that during the third time unit there are exactly $\frac{n}{2}$ free machines, we get that there are exactly $\frac{n}{2}$ tasks of Y and $\frac{n}{2}$ tasks of X executed in the second time unit. From the feasibility of the schedule we can conclude that these tasks form a balanced independent set $X_1' \cup Y_1'$ with $X_1' \subset X$, $Y_1' \subset Y$, and $|X_1'| = |Y_1'| = \frac{n}{2}$ in B, as required.

3. CONCLUDING REMARKS

We proved that the problem $P|\text{prec} = \text{bipartite}, p_j = 1|C_{\max} \leq 3$ is NP-complete. This problem is interesting to compare to $P|\text{prec}, p_j = 1, c_{ij} = 1|C_{\max} \leq 3$, which is polynomial [7], and to $P|\text{prec} = \text{bipartite}, p_j = 1, C_{ij} = 1|C_{\max} \leq 4$ which is NP-complete [4].

REFERENCES

[1] R. L. Graham, Bounds for Certain Multiprocessing Anomalies, *Bell System Tech. J.*, 1966, *45*, p. 1563-1581.

[2] M. R. Garey and D. S. Johnson, Computers and Intractability, A Guide to the Theory of NP-completeness, Ed. Freeman, 1979.

[3] R. L. Graham, E. L. Lawler, J. K. Lenstra and K. Rinnooy Kan, Optimization and Approximation in Deterministic Scheduling: A Survey, *Ann. Disc. Math.*, 1979, *5*, p. 287-326.

[4] J. A. Hoogeveen, J. K. Lenstra and B. Veltman, Three, four, five, six or the Complexity of Scheduling with Communication Delays, *Oper. Res. Lett.*, 1994, *16*, p. 129-137.

[5] J. K. Lenstra and A. H. G. Rinnooy Kan, Complexity of scheduling under precedence constraints, *Oper. Res.*, 1978, *26*, p. 22-35.

[6] C. H. Papadimitriou and K. Steiglitz "Combinatorial Optimization: Algorithms and Complexity", Prentice-Hall, 1982.

[7] C. Picouleau, Étude des problèmes d'optimisation dans les systèmes distribués, Ph.D. Thesis, Université Pierre-et-Marie-Curie, France, 1992.

[8] J. D. Ullman, NP-complete scheduling problems, *J. Comp. Syst. Sc.*, 1975, *10*, p. 384-393.

[9] R. Saad, Scheduling with Communication Delays, *J. Comb. Math. Comb. Comp.*, 1995, *18*, p. 214-224.