RAIRO. RECHERCHE OPÉRATIONNELLE

NELSON MACULAN
STELLA C. S. PORTO
CELSO C. RIBEIRO
CID CARVALHO DE SOUZA

A new formulation for scheduling unrelated processor under precedence constraints

RAIRO. Recherche opérationnelle, tome 33, n° 1 (1999), p. 87-92
<http://www.numdam.org/item?id=RO_1999__33_1_87_0>

RAIRO Rech. Opér.
(vol. 33, n° 1, 1999, pp. 87-92)

A NEW FORMULATION FOR SCHEDULING UNRELATED PROCESSOR UNDER PRECEDENCE CONSTRAINTS (*)

by Nelson MACULAN ([1]), Stella C. S. PORTO ([2]),
Celso C. RIBEIRO ([3]) and Cid CARVALHO DE SOUZA ([4])

Communicated by Catherine ROUCAIROL

Abstract. – *We give a new formulation for the problem of task scheduling into unrelated processors under precedence constraints. This formulation has a polynomial number of variables and does not require that the processing times be integer valued.*

Keywords: Parallel processing, scheduling, unrelated processors, precedence constraints, makespan.

Résumé. – *Nous donnons ici une formulation nouvelle du problème de l'ordonnancement des tâches utilisant des processeurs indépendants sous contraintes de cette formulation à un nombre polynomial de variables et n'exige pas que les temps de traitement aient des valeurs entières.*

Mots clés : Processus parallèles, ordonnancement, processeurs indépendants, contraintes de priorité.

1. MOTIVATION

Let $T = \{t_1, \ldots, t_n\}$ be a set of partially ordered tasks and $P = \{p_1, \ldots, p_m\}$ a set of processors. We are also given an (acyclic directed) precedence graph $G = [T, A]$ associated with the set of tasks [1, 4], such that $(t_i, t_j) \in A$ if and only if task t_i must be executed before task t_j. Each task has to be assigned to exactly one processor, in which it is entirely executed

(*) Received September 1996.

([1]) Universidade Federal do Rio de Janeiro COPPE – Engenharia de Sistemas e Computação Caixa Postal 68511, Rio de Janeiro 21945-970 Brazil, e-mail: maculan@cos.ufrj.br

([2]) Universidade Federal Fluminense Dept. of Telecommunication Engineering Rua Passos da Pátria 156, Niterói 24210, Brazil, e-mail: stella@caa.uff.br

([3]) Pontifícia Universidade Católica Departamento de Informática Rua Marquès de São Vicente 225, Rio de Janeiro 22453-900, Brazil e-mail: celso@inf.puc.rio.br

([4]) Universidade Estadual de Campinas Departamento de Ciência da Computação Caixa Postal 6065, Campinas 13081, Brazil, e-mail: cid@dcc.unicamp.br

Recherche opérationnelle/Operations Research, 0399-0559/99/01/$ 7.00

without preemption. For each task $t_j \in T$ and each processor $p_k \in P$, we denote by d_{jk} the total processing time of task t_j in case it is assigned to processor p_k. Three situations may occur:

- identical processors: $d_{j,k_1} = d_{j,k_2}$, $\forall t_j \in T$, $\forall p_{k_1}$, $p_{k_2} \in P$
- uniform processors: $d_{j_1,k_1}/d_{j_1,k_2} = d_{j_2,k_1}/d_{j_2,k_2}$, $\forall t_{j_1}$, $t_{j_2} \in T$, $\forall p_{k_1}$, $p_{k_2} \in P$
- unrelated processors: arbitrary processing times $d_{j,k}$ $\forall t_j \in T$, $\forall p_k \in P$.

The task scheduling problem under precedence constraints consists in finding an optimal assignment of the tasks in T to the processors in P minimizing the makespan. *i.e.* the maximum completion time among all tasks in T. The minimization of the makespan on two uniform processors (problem $Q2||C_{\max}$ in the notation of [8]) is already NP-hard [5, 6].

An application of this problem arises in the context of scheduling tasks of parallel programs. Parallel programs can be represented as a set of interrelated tasks which are sequential units. In a heterogeneous multiprocessor system, we not only have to determine how many, but also which processors should be allocated to an application, as well as which processors are going to be assigned to each task. Greedy algorithms for processor assignment of parallel applications modeled by task precedence graphs in heterogeneous multiprocessor architectures have been proposed by Menascé and Porto [9], while Porto and Ribeiro [11, 12] have studied sequential and parallel algorithms based on the tabu search metaheuristic.

The only known exact formulation [2] for this problem is due to Blazewicz *et al.* [3] and is based on the discretization of the schedule horizon into unit time-periods. Even regardless of more practical issues concerning its solvability as a large scale integer programming problem, this formulation has two main drawbacks. First, it requires that the processing times be integer valued. Second a non-polynomial number of 0-1 variables is used, due to the splitting of the schedule horizon into unit time-periods. Other works, such as Lasserre and Queyranne [7] and Queyranne and Schulz [13], deal with special cases or similar problems (such as single machine scheduling or multiple machine scheduling without precedence constraints) whose formulations cannot be extended to the more general problem considered in this paper.

In the next section we give a new formulation for the problem of task scheduling into unrelated processors under precedence constraints, involving only a polynomial number of 0-1 variables. Some final remarks are drawn in the last section.

2. FORMULATION WITH A POLYNOMIAL NUMBER OF VARIABLES

For ease of notation, we define the sets of indices $N = \{1, \ldots, n\}$ and $M = \{1, \ldots, m\}$ associated, respectively, with the sets T of tasks and P of processors. We define the following class of 0-1 variables:

$$y_{jk}^{s} = \begin{cases} 1, & \text{if task } t_j \text{ is the } s\text{-th task executed by processor } p_k, \\ 0, & \text{otherwise,} \end{cases}$$

for all $j \in N$, $k \in M$, and $s = 1, \ldots, n$. For every task $t_j \in T$, we denote by $w_j \geq 0$ the starting time of its execution. Moreover, we denote by $\Gamma(j)$ the set of the indices of the immediate predecessors of task t_j, *i.e.* $\Gamma(j) = \{i \in N | (t_i, t_j) \in A\}$. Then, the scheduling problem consisting in the minimization of the makespan $C_{\max}$ under precedence constraints may be formulated as:

$$\text{minimize } C_{\max} \tag{1}$$

subject to:

$$\sum_{k=1}^{m} \sum_{s=1}^{n} y_{jk}^{s} = 1, \quad \forall j \in N \tag{2}$$

$$\sum_{j=1}^{n} y_{jk}^{1} \leq 1, \quad \forall k \in M \tag{3}$$

$$\sum_{j=1}^{n} y_{jk}^{s} \leq \sum_{j=1}^{n} y_{jk}^{s-1}, \quad \forall k \in M, \quad \forall s = 2, \ldots, n \tag{4}$$

$$w_j \geq w_i + \sum_{k=1}^{m} \sum_{s=1}^{n} d_{ik} \cdot y_{ik}^{s}, \quad \forall i \in \Gamma(j), \quad \forall j \in N \tag{5}$$

$$w_j \geq w_i + d_{ik} - \alpha \cdot \left[2 - \left(y_{ik}^{s} + \sum_{r=s+1}^{n} y_{jk}^{r}\right)\right],$$
$$\forall k \in M, \forall s = 1, \ldots, n-1, \ \forall (i, j) \in N \times N \tag{6}$$

$$C_{\max} \geq w_j + \sum_{k=1}^{m} \sum_{s=1}^{n} d_{jk} \cdot y_{jk}^{s}, \quad \forall j \in N \tag{7}$$

$$y_{jk}^{s} \in \{0,1\}, \quad \forall j \in N, \quad \forall k \in M, \quad \forall s = 1, \ldots, n \tag{8}$$

$$w_j \geq 0, \quad \forall j \in N, \tag{9}$$

where $\alpha \gg 0$ is a sufficiently large penalty coefficient.

Equations (2) ensure that each task is assigned to exactly one processor. Inequalities (3-4) state that each processor can not be simultaneously used by more than one task. The first of those (3) means that at most one task will be the first one to be executed by any given processor $p_k \in P$. If a task is the s-th one (with $s \geq 2$) assigned to processor p_k, then there must be another one assigned as the $(s-1)$-th of this same processor, as stated by inequalities (4). Inequalities (5) express precedence constraints (no task may be started unless all its predecessors have already completed their execution), while constraints (7) define the maximum completion time, *i.e.* the makespan.

We now turn our attention to constraints (6), which define the sequence of starting times of the set of tasks assigned to the same processor. They express the fact that task t_j must start at least d_{ik} time units after the beginning of task t_i, whenever it is executed after task t_i in the same processor p_k, *i.e.* $y_{ik}^{s} = \sum_{r=s+1}^{n} y_{jk}^{r} = 1$ for some $s = 1, \ldots, n-1$.

3. FINAL REMARKS

The success of integer programming methods is known to be very dependent on the bound given by the linear relaxation provided by the formulation. To improve this bound, we seek for stronger inequalities to add to the model. The formulation given in the previous section may be improved further.

One way to measure the "strength" of an inequality is to compute the amount of (affine independent) integer solutions satisfying it at equality. The larger this amount is, the stronger is the inequality (see [10] for details). Inequalities (6) may be strengthened, in such as way that the number of integer solutions satisfying these inequalities at equality increases. To do so, we add new terms to these inequalities, in a process usually called lifting. Assuming that i, j, k, and s are fixed, we notice that for a solution to

verify inequality (6) at equality, tasks t_i and t_j have to be respectively the s-th and $(s+1)$-th tasks executed by processor p_k and, moreover, this processor cannot remain idle between the execution of these two tasks. Lifting inequalities (6) leads to the stronger valid inequalities (6′) below:

$$w_j \geq w_i + d_{ik} - \alpha \cdot \left[2 - \left(\sum_{r=1}^{s} y_{ik}^r + \sum_{r=s+1}^{n} y_{jk}^r\right)\right] + \sum_{l \neq i \neq j,\, l=1}^{n} d_{lk} \left(y_{lk}^s + y_{jk}^{s+1} + \sum_{r=s+1}^{n} y_{jk}^r - 1\right) \quad (6')$$
$$\forall k \in M, \quad s = 1, \ldots, n-1, \quad \forall (i, j) \in N \times N.$$

We do not claim that the original formulation or the new one are directly amenable to be solved by standard integer programming techniques, such as branch-and-bound or branch-and-cut. However, as already mentionned, the original formulation already has two main drawbacks. First, it requires that the processing times be integer valued. Second, a non-polynomial number of 0-1 variables is used, due to the splitting of the schedule horizon into unit time-periods. Both of these difficulties are overcam by the new formulation.

We have developed in this paper a new, more suitable formulation for the problem of task scheduling into unrelated processors under precedence constraints. This formulation makes use of only a polynomial number of 0-1 variables. Moreover, the assumption of integer valued processing times is no longer necessary. Current research and further extensions of this work consist in (1) finding good relaxations of this formulation, which could generate sharper lower bounds for the optimal value of the makespan; (2) solving practical applications on regularly structured precedence graphs, corresponding to real-life processor configurations; (3) deriving preprocessing procedures for variable fixation and coefficient reduction; and (4) its application to the problems already solved by Porto and Ribeiro [11, 12], which would allow the evaluation of the quality of the approximate solutions given by tabu search.

REFERENCES

1. D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation*, Prentice-Hall, Englewood Cliffs, 1989.
2. J. Blazewicz, *Personnal communication*, 1993.
3. J. Blazewicz, W. Cellary, R. Slowinski and J. Weglarz, *Scheduling Under Resource Constraints – Deterministic Models,* J.C. Baltzer AG, Basel, 1986.

4. E. G. COFFMAN and P. J. DENNING, Operating Systems Theory, Prentice-Hall Inc., New Jersey, 1973.
5. M. R. GAREY and D. S. JOHNSON, Strong NP-Completeness Results: Motivation, Examples and Implications, *Journal of the ACM* 25, 1978, pp. 499-508.
6. M. R. GAREY and D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* W. H. Freeman and Company, San Francisco, 1979.
7. J.-B. LASSERRE and M. QUEYRANNE, Generic Scheduling Polyhedra and a New Mixed-Integer Formulation for Single Machine Scheduling, in *Proceedings of the II Integer Programming and Combinatorial Optimization Conference* (E. Balas, G. Cornuejols, and R. Kannan, eds.), pp. 136-149.
8. E. L. LAWLER, J. K. LENSTRA, A. H. G. RINNOOY KAN and D. B. SHMOYS, Sequencing and Scheduling: Algorithms and Complexity, Report NFI 11.89/03, Eindhoven Institute of Technology, Department of Mathematics and Computer Science, Eindhoven, 1989.
9. D. A. MENASCÉ and S. C. S. PORTO, Processor Assignment in Heterogeneous Parallel Architectures, *Proceedings of the IEEE International Parallel Processing Symposium,* pp. 186-191, Beverly Hills, 1992.
10. G. L. NEMHAUSER and L. A. WOLSEY, *Integer and Combinatorial Optimization,* John Wiley and Sons, 1988.
11. S. C. PORTO and C. C. RIBEIRO, A Tabu Search Approach to Task Scheduling on Heterogeneous Processors under Precedence Constraints, *International Journal of High Speed Computing* 7, 1995, pp. 45-71.
12. S. C. PORTO and C. C. RIBEIRO, Parallel Tabu Search Message-Passing Synchronous Strategies for Task Scheduling under Precedence Constraints, *Journal of Heuristics* 1, 1995, pp. 207-223.
13. M. QUEYRANNE and A. SCHULZ, Polyhedral Approaches to Machine Scheduling, Report 1994-408, Technical University of Berlin, 1994.