Multiprocessor Scheduling Problem with Stepwise Model of Job Value Change

Adam Janiak and Tomasz Krysiak

Institute of Engineering Cybernetics, Wroclaw University of Technology,
Janiszewskiego 11/17, 50-372 Wrocław, Poland
Adam.Janiak@pwr.wroc.pl, Tomasz.Krysiak@pwr.wroc.pl

Abstract. The paper deals with a scheduling problem on the parallel processors, in which the sum of values of all the jobs is maximized. The value of job is characterized by a stepwise non-increasing function. Establishing an order of processing of datagrams which are sent by multiprocessor router is a practical example of application of this problem. It was already proved that its single processor case is NP-hard, thus the problem is also NP-hard. Therefore, two pseudo-polynomial time algorithms for the problem with common moments of job value change and a polynomial time algorithm for the case with identical processing times are constructed. It is also constructed and experimentally tested a number of heuristic algorithms which solve the general version of the problem.

1 Introduction

The paper deals with a scheduling problem on the parallel processors, in which the sum of values of all the jobs is maximized and a job value is described by a non-increasing stepwise function. Establishing an order of processing of datagrams which are sent by router is an application example of the problem, if a multiprocessor router is given. The precise description of this situation (but not for multiprocessor case) is given in [2], since a single processor case of this problem was considered there. It was proved in the mentioned paper, that the problem is NP-hard. Moreover, a pseudo-polynomial time algorithm for some special case, and a number of heuristic algorithms for general single-processor version of the problem, were constructed. In this paper, based on the results presented in [2], we construct two pseudo-polynomial time algorithms and some heuristics for parallel processor cases of the problem. Thus, this paper is an extension of [2].

The remaining part of the paper is organized as follows. In the next section we formulate the problem and next we prove polynomially solvable cases of the problem. Section 4 deals with the pseudo-polynomial time algorithms constructed for the cases with common moments of job value change and in Section 5 we present and experimentally compare some heuristic algorithms constructed to solve the general version of the problem. Some concluding remarks are given in Section 6.

2 Problem Formulation

There are given a set of m identical processors $M = \{M_1, \ldots, M_m\}$ and a set of n independent and non-preemptive jobs $J = \{J_1, \ldots, J_n\}$, immediately available for processing at time 0. Each processor can execute at the most one job at a time. Whereas, each job i is characterized by its processing time $p_i > 0$, its value $v_i(t)$ dependent on the time t and the moments $d_{ij} > 0$, $j = 1 \ldots k-1$ at which a change of job value occur. The model of job value is given by non-increasing stepwise function defined as follows

$$v_i(t) = \begin{cases} w_{i1}, & 0 < t \le d_{i1} \\ w_{i2}, & d_{i1} < t \le d_{i2} \\ \vdots & \\ w_{ik}, & d_{ik-1} < t \end{cases},$$

where $w_{i1} > w_{i2} > \ldots > w_{ik}$.

Solution of the problem is represented by a set of permutations $\Pi = \{\pi_1, \ldots, \pi_l, \ldots, \pi_m\}$, where π_l denotes a schedule of jobs assigned to execute on a processor M_l. Let n_l denotes a number of jobs processed on M_l (then $\sum_{l=1}^{m} n_l = n$). The objective is to find such an assignment of jobs to the processors and such a schedule of jobs on the processors $\Pi = \{\pi_1, \ldots, \pi_m\}$, for which the sum of job values, calculated at their completion times $C_{\pi_l(i)}$ (if a job is processed on the i^{th} position of the permutation π_l), is *maximal*:

$$\sum_{l=1}^{m} \sum_{i=1}^{n_l} v_{\pi_l(i)}(C_{\pi_l(i)}) \Rightarrow \max .$$

Therefore, using the three-field notation $\alpha \mid \beta \mid \gamma$ for scheduling problems [1], the problem considered in the paper is given by

$$P \left| v_i(C_i) = \begin{cases} w_{i1}, & 0 < C_i \le d_{i1} \\ w_{i2}, & d_{i1} < C_i \le d_{i2} \\ \vdots & \\ w_{ik}, & d_{ik-1} < C_i \end{cases} \right| \sum v_i(C_i).$$

3 Polynomially Solvable Cases

Property 1 *Problem $P\,|p_i = p|\sum v_i(C_i)$, where $v_i(C_i)$ are some arbitrary functions, can be solved optimally in $O(n^3)$ time.*

Proof. Since $p_i = p$, thus there exist exactly n moments $t_j = \lceil \frac{j}{m} \rceil \cdot p$ $(j = 1, \ldots, n)$, at which the jobs complete. Therefore, the problem $P\,|p_i = p|\sum v_i(C_i)$ is solved by finding an optimal assignment of the jobs to these moments. This amounts to formulating and solving an $n \times n$ usual maximization assignment problem with cost coefficients $c_{ij} = v_i(t_j)$. It is known that the assignment problem can be solved in $O(n^3)$ steps ([3]). $\square$