General Multiprocessor Task Scheduling: Approximate Solutions in Linear Time *

Klaus Jansen¹ and Lorant Porkolab²

¹ IDSIA Lugano, Corso Elvezia 36, 6900 Lugano, Switzerland
   klaus@idsia.ch
² Max-Planck-Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany
   porkolab@mpi-sb.mpg.de

Abstract. We study the problem of scheduling a set of $n$ independent tasks on a fixed number of parallel processors, where the execution time of a task is a function of the subset of processors assigned to the task. We propose a fully polynomial approximation scheme that for any fixed $\epsilon > 0$ finds a preemptive schedule of length at most $(1 + \epsilon)$ times the optimum in $O(n)$ time. We also discuss the non-preemptive variant of the problem, and present a polynomial approximation scheme that computes an approximate solution of any fixed accuracy in linear time. In terms of the running time, this linear complexity bound gives a substantial improvement of the best previously known polynomial bound [5].

1 Introduction

In classical scheduling theory, each task is processed by only one processor at a time. However recently, due to the rapid development of parallel computer systems, new theoretical approaches have emerged to model scheduling on parallel architectures. One of these is scheduling multiprocessor tasks, see e.g. [3, 6, 7].

In this paper we address some multiprocessor scheduling problems, where a set of $n$ tasks has to be executed by $m$ processors such that each processor can work on at most one task at a time and a task can (or may need to be) processed simultaneously by several processors. In the dedicated variant of this model, each task requires the simultaneous use of a prespecified set of processors. A generalization of the dedicated variant allows tasks to have a number of alternative modes, where each processing mode specifies a subset of processors and the task’s execution time on that particular processor set. This problem is called general multiprocessor task scheduling.

Depending on the model, tasks can be preempted or not. In the non-preemptive model, a task once started has to be completed without interruption. In the preemptive model, each task can be interrupted any time at no cost and restarted later possibly on a different set of processors. The objective of the scheduling problems discussed in this paper is to minimize the makespan, i.e. the maximum

* Supported by EU ESPRIT LTR Project No. 20244 (ALCOM-IT) and by the Swiss Office Fédéral de l’éducation et de la Science project n 97.0315 titled "Platform".
completion time $C_{\text{max}}$. The dedicated and general variants of non-preemptive (preemptive) scheduling for independent multiprocessor tasks on a fixed number of processors are denoted by $Pm|\text{fix}_i|C_{\text{max}}$ ($Pm|\text{fix}_j,\text{pmtn}|C_{\text{max}}$) and $Pm|\text{set}_j|C_{\text{max}}$ ($Pm|\text{set}_j,\text{pmtn}|C_{\text{max}}$), respectively.

Regarding the complexity, problems $P3|\text{fix}_i|C_{\text{max}}$ and $P3|\text{set}_j|C_{\text{max}}$ are strongly NP-hard [9], thus they do not have fully polynomial approximation schemes, unless P=NP. Recently, Amoura et al. [1] developed a polynomial time approximation scheme for $Pm|\text{fix}_j|C_{\text{max}}$. For the general problem $Pm|\text{set}_j|C_{\text{max}}$, Bianco et al. [2] presented an approximation algorithm whose approximation ratio is bounded by $m$. Later Chen and Lee [4] improved their algorithm by achieving an approximation ratio $\frac{m}{2} + \epsilon$. Until very recently, this was the best approximation result for the problem, and it was not known whether there is a polynomial-time approximation scheme or even a polynomial-time approximation algorithm with an absolute constant approximation guarantee. Independently from our work presented here, Chen and Miranda [5] have recently proposed a polynomial-time approximation scheme for the problem. The running time of their approximation scheme is $O(n^{\lambda_{m,\epsilon} + j_{m,\epsilon} + 1})$, where $\lambda_{m,\epsilon} = (2j_{m,\epsilon} + 1)B_m$ and $j_{m,\epsilon} \leq (3mB_m + 1)^{m/\epsilon}$ with $B_m \leq m!$ denoting the $m$th Bell number. In this paper, we propose another polynomial time approximation scheme for $Pm|\text{set}_j|C_{\text{max}}$ that computes an $\epsilon$-approximate solution in $O(n)$ time for any fixed positive accuracy $\epsilon$. This gives a substantial improvement - in terms of the running time - of the previously mentioned result [5], and also answers an open question of the latter paper by providing an approach that is not based on dynamic programming.

It is known that the preemptive variant $Pm|\text{set}_j,\text{pmtn}|C_{\text{max}}$ of the problem can be solved in polynomial time [2] by formulating it as a linear program with $n$ constraints and $n^m$ variables and computing an optimal solution by using any polynomial-time linear programming algorithm. Even though (for any fixed $m$), the running time in this approach is polynomial in $n$, the degree of this polynomial depends linearly on $m$. Therefore it is natural to ask whether there are more efficient algorithms for $Pm|\text{set}_j,\text{pmtn}|C_{\text{max}}$ (of running time, say for instance, $O(n)$ or $O(n^c)$ with an absolute constant $c$) that compute exact or approximate solutions. In this paper we focus on approximate solutions and present a fully polynomial approximation scheme for $Pm|\text{set}_j,\text{pmtn}|C_{\text{max}}$ that finds an $\epsilon$-approximate solution in $O(n)$ time for any fixed positive accuracy $\epsilon$. This result also shows that as long as approximate solutions (of any fixed positive accuracy) are concerned, linear running time can be achieved for the problem; but leaves open the question (which also partly motivated this work) whether there exists a linear-time algorithm for computing an exact optimal solution of $Pm|\text{set}_j,\text{pmtn}|C_{\text{max}}$. This question was answered in an affirmative way for the dedicated variant $Pm|\text{fix}_j,\text{pmtn}|C_{\text{max}}$ of the problem by Amoura et al. [1].