On Preemptive Resource Constrained Scheduling: Polynomial-Time Approximation Schemes*

Klaus Jansen[1] and Lorant Porkolab[2]

[1] Institut für Informatik und praktische Mathematik, Christian-Albrechts-Universität zu Kiel, 24 098 Kiel, Germany
kj@informatik.uni-kiel.de

[2] Applied Decision Analysis, PricewaterhouseCoopers London, United Kingdom
lorant.porkolab@uk.pwcglobal.com

Abstract. We study resource constrained scheduling problems where the objective is to compute feasible preemptive schedules minimizing the makespan and using no more resources than what are available. We present approximation schemes along with some inapproximibility results showing how the approximability of the problem changes in terms of the number of resources. The results are based on linear programming formulations (though with exponentially many variables) and some interesting connections between resource constrained scheduling and (multi-dimensional, multiple-choice, and cardinality constrained) variants of the classical knapsack problem. In order to prove the results we generalize a method by Grigoriadis et al. for the max-min resource sharing problem to the case with weak approximate block solvers (i.e. with only constant, logarithmic, or even worse approximation ratios). Finally we present applications of the above results in fractional graph coloring and multiprocessor task scheduling.

1 Introduction

In this paper we consider the general preemptive resource constrained scheduling problem denoted by $P|res...,pmtn|C_{\max}$: There are given n tasks $\mathcal{T} = \{T_1, \ldots, T_n\}$, m identical machines and s resources such that each task $T_j \in \mathcal{T}$, is processed by one machine requiring p_j units of time and r_{ij} units of resource i, $i = 1, \ldots, s$, from which only c_i units are available at each time. One may assume w.l.o.g. that $r_{ij} \in [0, 1]$ and $c_i \geq 1$. The objective is to compute a preemptive schedule of the tasks minimizing the maximum completion time $C_{\max}$. The three dots in the notation indicate that there are no restrictions on the number of resources s, the largest possible capacity o and resource requirement r values, respectively. If any of these is limited, the corresponding fixed limit replaces the corresponding dot in the notation (e.g. if $s \leq 1$, then $P|res1..,pmtn|C_{\max}$ is used, or if $r_{ij} \leq r$, then $P|res..r,pmtn|C_{\max}$). We will study different variants of the

* Supported in part by EU Thematic Network APPOL I + II, Approximation and Online Algorithms, IST-1999-14084 and IST-2001-30012 and by the EU Research Training Network ARACNE, Approximation and Randomized Algorithms in Communication Networks, HPRN-CT-1999-00112.

W.J. Cook and A.S. Schulz (Eds.): IPCO 2002, LNCS 2337, pp. 329–349, 2002.

problem and their applications in multiprocessor task scheduling and fractional graph coloring.

1.1 Related Results

Resource constrained scheduling is one of the classical scheduling problems. Garey and Graham [12] proposed approximation algorithms for the non-preemptive variant $P|res...|C_{\max}$ with approximation ratios $s+1$ (when the number of machines is unbounded, $m \geq n$) and $\min(\frac{m+1}{2}, s+2-\frac{2s+1}{m})$ (when $m \geq 2$). Further results are known for some special cases: Garey et al. [13] proved that when $m \geq n$ and each task T_j has unit-execution time, i.e. $p_j = 1$, the problem (denoted by $P|res..., p_j = 1|C_{\max}$) can be solved by First Fit and First Fit Decreasing heuristics providing asymptotic approximation ratio $s + \frac{7}{10}$ and a ratio between $s + ((s-1)/s(s+1))$ and $s + \frac{1}{3}$, respectively. De la Vega and Lueker [6] gave a linear-time algorithm with asymptotic approximation ratio $s+\epsilon$ for each fixed $\epsilon > 0$. Further results and improvements for non-preemptive variant are given in [5,38,39].

For the preemptive variant substantially less results are known: Blazewicz et al. [2] proved that when m is fixed, the problem $Pm|pmtn, res...|C_{\max}$ (with identical machines) and even the variant $Rm|pmtn, res...|C_{\max}$ (with unrelated machines) can be solved in polynomial time. Krause et al. [25] studied $P|pmtn, res1..|C_{\max}$, i.e. where there is only one resource ($s = 1$) and proved that both First Fit and First Fit Decreasing heuristics can guarantee $3 - 3/n$ asymptotic approximation ratio.

A related problem is multiprocessor task scheduling, where a set $\mathcal{T}$ of n tasks has to be executed by m processors such that each processor can execute at most one task at a time and each task must be processed by several processors in parallel. In the parallel (non-malleable) model $P|size_j|C_{\max}$, there is a value $size_j \in M = \{1, \ldots, m\}$ given for each task T_j indicating that T_j can be processed on any subset of processors of cardinality $size_j$ [1,7,8,18,23]. In the malleable variant $P|fctn|C_{\max}$, each task can be executed on an arbitrary subset of processors, and the execution time $p_j(\ell)$ depends on the number ℓ of processors assigned to it [27,31,40]. Regarding the complexity, it is known [7,8] that the preemptive variant $P|size_j, pmtn|C_{\max}$ is NP-hard. In [20], focusing on computing optimal solutions, we presented an algorithm for solving the problem $P|size_j, pmtn|C_{\max}$ and showed that this algorithm runs in $O(n) + poly(m)$ time, where $poly(.)$ is a univariate polynomial. Furthermore, we extended this algorithm also to malleable tasks with running time polynomial in m and n. These results are based on methods by Grötschel et al. [17] and use the ellipsoid method.

Another related problem is fractional graph coloring, see e.g. [10,16,28,30,32,35,36]. Grötschel et al. [16] proved that the weighted fractional coloring problem is NP-hard for general graphs, but can be solved in polynomial time for perfect graphs. They have proved the following interesting result: For any graph class $\mathcal{G}$, if the problem of computing $\alpha(G, w)$ (the weight of the largest weighted independent set in G) for graphs $G \in \mathcal{G}$ is NP-hard, then the problem of determining the