ARTICLE IN PRESS
JID: EOR [m5G;September 14, 2016;5:54]

European Journal of Operational Research 000 (2016) 1–11

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Approximation schemes for parallel machine scheduling with non-renewable resources

Péter Györgyi[a,b], Tamás Kis[b,*]

[a] Department of Operations Research, Loránd Eötvös University, H1117 Budapest, Pázmány Péter sétány 1/C, Hungary
[b] Institute for Computer Science and Control, H1111 Budapest, Kende str. 13–17, Hungary

ARTICLE INFO

Article history:
Received 13 August 2015
Accepted 5 September 2016
Available online xxx

Keywords:
Scheduling
Parallel machines
Non-renewable resources
Approximation schemes

ABSTRACT

In this paper the approximability of parallel machine scheduling problems with resource consuming jobs is studied. In these problems, in addition to a parallel machine environment, there are non-renewable resources, like raw materials, energy, or money, consumed by the jobs. Each resource has an initial stock, and some additional supplies at a-priori known moments in time and in known quantities. The schedules must respect the resource constraints as well. The optimization objective is either the makespan, or the maximum lateness. Polynomial time approximation schemes are provided under various assumptions, and it is shown that the makespan minimization problem is APX-complete if the number of machines is part of the input even if there are only two resources.

1. Introduction

In Supply Chains, non-renewable resources like raw materials, or energy are taken into account from the design through the operational levels. Advanced planning systems explicitly model and optimize their usage at various planning levels, see e.g., Chapters 4, 9 and 10 of Stadtler and Kilger (2008). In this paper, we focus on short-term scheduling, where in addition to machines, there are non-renewable resources consumed by the jobs. Each non-renewable resource has an initial stock, which is replenished at a-priori known moments of time and in known quantities.

More formally, there are m parallel machines, $\mathcal{M} = \{M_1, \ldots, M_m\}$, a finite set of n jobs $\mathcal{J} = \{J_1, \ldots, J_n\}$, and a finite set of non-renewable resources $\mathcal{R}$ consumed by the jobs. Each job J_j has a processing time $p_j \in \mathbb{Z}_+$, a release date r_j, and resource requirements $a_{ij} \in \mathbb{Z}_+$ from the resources $i \in \mathcal{R}$. Preemption of jobs is not allowed and each machine can process at most one job at a time. The resources are supplied in q different time moments, $0 = u_1 < u_2 < \cdots < u_q$; the vector $\tilde{b}_\ell \in \mathbb{Z}_+^{|\mathcal{R}|}$ represents the quantities supplied at u_ℓ. A *schedule* σ specifies a machine and the starting time S_j of each job and it is *feasible* if (i) on every machine the jobs do not overlap in time, (ii) $S_j \ge r_j$ for each $j \in \mathcal{J}$, and if (iii) at any time point t the total supply from each resource is at least the total request of those jobs starting not later than t, i.e., $\sum_{(\ell\,:\,u_\ell \le t)} \tilde{b}_{\ell i} \ge \sum_{(j\,:\,S_j \le t)} a_{ij}$, $\forall i \in \mathcal{R}$. We will consider two types of objective functions: the minimization of the maximum job completion time (makespan) defined by $C_{\max} = \max_{j \in J} C_j$; and the minimization of the maximum lateness, i.e., each job has a due-date d_j, $j \in \mathcal{J}$, and $L_{\max} := \max_{j \in \mathcal{J}}(C_j - d_j)$. Clearly, $L_{\max}$ is a generalization of $C_{\max}$.

Assumption 1. $\sum_{\ell=1}^{q} \tilde{b}_{\ell i} = \sum_{j \in \mathcal{J}} a_{ij}$, $\forall i \in \mathcal{R}$, holds without loss of generality.

Since the makespan minimization problem with resource consuming jobs on a single machine is NP-hard even if there are only two supply dates (Carlier, 1984), all problems studied in this paper are NP-hard.

Scheduling with non-renewable resources has a great practical interest. Chapter 4 of Stadtler and Kilger (2008) describes examples in consumer goods industry and in computer assembly, where purchased items have to be taken into account at several planning levels including short-term scheduling which is the topic of the present paper. Herr and Goel (2016) study a scheduling problem arising in the continuous casting stage of steel production. A continuous caster is fed with ladles of liquid steel, where each ladle contains a certain steel grade and has orders allocated to it that determine a due date. The liquid steel is produced from hot iron supplied by a blast furnace with a constant rate. The sequence of ladles, including setups between ladles of different setup families, is not allowed to consume more hot metal than supplied by the blast furnace. Belkaid, Maliki, Boudahri, and Sari (2012) study a problem of order picking in a platform with a

* Corresponding author.
E-mail addresses: gyorgyi.peter@sztaki.mta.hu (P. Györgyi), kis.tamas@sztaki.mta.hu, kis.tamas@yahoo.com (T. Kis).

http://dx.doi.org/10.1016/j.ejor.2016.09.007
0377-2217/

Please cite this article as: P. Györgyi, T. Kis, Approximation schemes for parallel machine scheduling with non-renewable resources, European Journal of Operational Research (2016), http://dx.doi.org/10.1016/j.ejor.2016.09.007

distribution company that leads to the model considered in this paper. In Carrera, Ramdane-Cherif, and Portmann (2010), a similar problem is investigated in a shoe-firm. Further applications can be found in Section 2.

In this paper we take a theoretical viewpoint and analyze the approximability of parallel machine scheduling problems augmented with non-renewable resources. We believe that our study leads to a deeper understanding of the problem, that may facilitate the development of efficient practical algorithms.

1.1. Terminology

An *optimization problem* Π consists of a set of instances, where each instance has a set of *feasible solutions*, and each solution has an (objective function) value. In a *minimization problem* a feasible solution of minimum value is sought, while in a maximization problem one of maximum value. An ε-*approximation algorithm* for an optimization problem Π delivers in polynomial time for each instance of Π a solution whose objective function value is at most $(1+\varepsilon)$ times the optimum value in case of minimization problems, and at least $(1-\varepsilon)$ times the optimum in case of maximization problems. For an optimization problem Π, a family of approximation algorithms $\{A_\varepsilon\}_{\varepsilon>0}$, where each A_ε is an ε-approximation algorithm for Π is called a *Polynomial Time Approximation Scheme (PTAS) for* Π.

Observation 1. For a PTAS for some problem Π, it is sufficient to provide a family of algorithms $\{A_\varepsilon\}_{\varepsilon>0}$ where each A_ε is an $c\cdot\varepsilon$-approximation algorithm for Π, where the constant factor c does not depend on the input or on ε. Then, letting $\varepsilon := \delta/c$, we get a PTAS $\{A_{(\delta/c)}\}_{\delta>0}$ for Π.

We use the standard $\alpha|\beta|\gamma$ notation for scheduling problems (Graham, Lawler, Lenstra, & Kan, 1979), where α denotes the processing environment, β the additional restrictions, and γ the objective function. In this paper, $\alpha = Pm$, which indicates m parallel machines for some fixed m. In the β field, 'rm' means that there are non-renewable resource constraints, $rm = r$ indicates $|\mathcal{R}| = r$. Further options are $q = const$ meaning that the number of supplies is a fixed constant, r_j indicates job release dates, while the restriction $\#\{r_j : r_j < u_q\} \leq const$ bounds the number of distinct job release dates before the last supply date u_q by a constant. For a set H, we define $p(H) := \sum_{j\in H} p_j$.

Throughout the paper we will consider *monotone* objective functions $F_{\max}$ that satisfy the following conditions:

(i) $F_{\max}$ is monotone increasing in the job completion times, i.e., $F_{\max}(C_1,\ldots,C_n) \leq F_{\max}(C'_1,\ldots,C'_n)$, for arbitrary $0 \leq C_j \leq C'_j$, $j = 1,\ldots,n$,
(ii) Its value does not grow faster than the value of any of its arguments, i.e., $F_{\max}(C_1+\delta,\ldots,C_n+\delta) \leq F_{\max}(C_1,\ldots,C_n)+\delta$ for any $\delta \geq 0$,
(iii) On any instance, and for any feasible schedule, $F_{\max}$ is at least u_q.

Notice that e.g., the makespan, and the maximum lateness increased by some (instance dependent) constant satisfy the above properties, but the total completion time does not. From now on $F_{\max}$ denotes an arbitrary monotone objective function.

1.2. Main results

If the number of the machines is part of the input, then we have the following non-approximability result:

Theorem 1. *Deciding whether there is a schedule of makespan 2 with two non-renewable resources, two supply dates and unit-time jobs on an arbitrary number of machines* ($P|rm=2, q=2, p_j=1|C_{\max}\leq 2$) *is NP-hard.*

Corollary 1. *It is NP-hard to approximate problem* $P|rm=2, q=2, p_j=1|C_{\max} \leq 2$ *better than* $3/2-\varepsilon$ *for any* $\varepsilon > 0$.

By assumption 1, the optimum makespan is at least u_q, therefore, a straightforward two-approximation algorithm would schedule all the jobs after u_q. Therefore, we have the following result.

Corollary 2. $P|rm=2, q=2, p_j=1|C_{\max}$ *is APX-complete.*

The following result helps to obtain polynomial time approximation schemes for the general problem $P[m]|rm, r_j|F_{\max}$, provided that we have a family of approximation algorithms for restricted versions of the problem.

Proposition 1. *In order to have a PTAS for* $P[m]|rm, r_j|F_{\max}$, *it suffices to provide a family of algorithms* $\{A_\varepsilon\}_{\varepsilon>0}$ *such that* A_ε *is an* ε*-approximation algorithm for the restricted problem where the supply dates and the job release dates before* u_q *are from the set* $\{\ell\varepsilon u_q : \ell = 0,1,2,\ldots,\lfloor 1/\varepsilon\rfloor\}$.

Using Proposition 1, we can prove the following result:

Theorem 2. $Pm|rm = const., r_j|C_{\max}$ *admits a PTAS.*

Notice that a PTAS has been known only for $1|rm = const, q = const, \#\{r_j : r_j < u_q\} \leq const|C_{\max}$ (Györgyi & Kis, 2015b). If the jobs are dedicated to machines, we have an analogous statement:

Theorem 3. $Pm|rm = const., r_j, ddc|C_{\max}$ *admits a PTAS.*

Now we turn to the $L_{\max}$ objective. Since the optimum lateness may be 0 or negative, a standard trick is to increase the lateness of the jobs by a constant that depends on the input. In our case, let $L'_{\max} := \max_j\{C_j - d_j + D\}$, where $D := \max_{j\in\mathcal{J}}\{d_j\} + u_q$. Note that this function satisfies the conditions (i)–(iii), thus it is a monotone objective function. In order to provide a PTAS for the lateness objective, we have to assume that the processing times are proportional to the resource consumptions. Such a model with the makespan objective has already been studied in Györgyi and Kis (2015b).

Theorem 4. *If* $L'_{\max}$ *is defined as above, then* $Pm|rm = 1, p_j = a_j|L'_{\max}$ *admits a PTAS.*

In Table 1 we summarize known and new approximability results for scheduling resource consuming jobs in single machine as well as in parallel machine environments, when preemption of processing is not allowed, and the resources are consumed right at starting the jobs. The table contains results for the makespan, the maximum lateness, and the weighted completion time objectives. These results complement the large body of approximation algorithms for NP-hard single and parallel machine scheduling problems (Williamson & Shmoys, 2011).

1.3. Structure of the paper

In Section 2 we summarize previous work on machine scheduling with non-renewable resources. In Section 3 we prove our hardness result Theorem 1. Then in Section 4 we establish Proposition 1. In Sections 5.2, 6, and 7 we prove Theorems 2, 3, and 4, respectively. Finally, we conclude the paper in Section 8.

2. Previous work

Scheduling problems with resource consuming jobs were introduced by Carlier (1984), Carlier and Rinnooy Kan (1982), and Slowinski (1984). In Carlier (1984), the computational complexity of several variants with a single machine was established, while in