Operations Research Letters 51 (2023) 421–424

Contents lists available at ScienceDirect

Operations Research Letters

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

An FPTAS for scheduling with resource constraints

Bo Chen [a,*], Vitaly A. Strusevich [b]

[a] Warwick Business School, The University of Warwick, Coventry, CV4 7AL, UK
[b] Sherwood Road, Welling, Kent, UK

ARTICLE INFO

Article history:
Available online 14 June 2023

Keywords:
Scheduling
Approximation
Resource constraint

ABSTRACT

We study a basic scheduling problem with resource constraints: A number of jobs need to be scheduled on two parallel identical machines with the objective of minimizing the makespan, subject to the constraint that jobs may require a unit of one of the given renewable resources during their execution. For this NP-hard problem, we develop a fully polynomial-time approximation scheme (FPTAS). Our FPTAS makes a novel use of existing algorithms for the subset-sum problem and the open shop scheduling problem.

1. Introduction

Scheduling subject to resource constraints is common practice, particularly in the context of project management and computer operating systems. In this note we consider a very basic model of scheduling jobs on two parallel identical machines subject to some simple resource constraint, in which each job requires up to one unit of one of the given renewable and non-divisible resources during its execution. Such a model can find applications in satellite data download management [6], in management of photolithography equipment in the microelectronic industry [7] and in human resource management [13].

In order to denote scheduling problems in a clear and compact way, the three-field classification scheme of the form $\alpha|\beta|\gamma$ has become standard, where α represents a machine environment, β describes the processing conditions and γ is the objective function. For example, the classical problem of minimizing the makespan $C_{\max}$ (i.e., the maximum completion time) on m parallel identical machines is denoted either by $P|\,|C_{\max}$ if the number of machines is variable (part of the input) or by $Pm|\,|C_{\max}$ if the number of machines is fixed at m. The empty middle field means that all processing conditions are set at default, including that interruption of execution of any job is not allowed. A widely accepted notation for resource constraints as part of processing conditions is introduced in [1] and also considered in [2,4]. Here we will use a slightly extended version as presented in [13], in which a string of the form "res $\rho_1\rho_2\rho_3\rho_4$" is added to the middle field β, where

- ρ_1 is the number of available renewable resources;
- ρ_2 is an upper bound on the number of resources a job may need;
- ρ_3 is an upper bound on the number of units of any resource available at a time;
- ρ_4 is an upper bound on the number of units of any resource that can be consumed by a job at a time.

The value of each of these parameters is either a known constant or the symbol "·" if the parameter is variable (i.e., part of the input). With the extended notation, the problem of our primary concern is denoted by $P2|\text{res}\cdot 111|C_{\max}$. In other words, we are concerned with scheduling jobs on two parallel identical machines to minimize the makespan subject to the constraint that each job needs either none or one unit of one of a number of given renewable resources.

The known NP-hardness of problem $P2|\,|C_{\max}$ motivates our study on the approximability of the problem at hand. In this note we answer the question whether problem $P2|\text{res}\cdot 111|C_{\max}$ admits an FPTAS by providing an FPTAS for the problem with running time $O(n/\varepsilon)$, the same time complexity as the FPTAS for problem $P2|\,|C_{\max}$ (with no resource constraints).

2. Literature and preliminaries

Our model can be formally described as follows. Each job of a set $N=\{1,\dots,n\}$ has to be processed without interruption on one of two identical machines, M_1 and M_2. There are q renewable resources and exactly one unit of each of these resources is available at any time. A job in N may need one unit of one of these q resources during its execution. If we denote by N^k the set of jobs that require resource k $(1\le k\le q)$ and by N^0 the set of jobs that require no resources, then set N is partitioned into $N^0, N^1, \dots, N^q$.

* Corresponding author.
E-mail address: B.Chen@warwick.ac.uk (B. Chen).

https://doi.org/10.1016/j.orl.2023.06.001

For any job $j \in N$, denote its processing time by $p(j)$. For any subset $U \subseteq N$ of jobs denote $p(U) = \sum_{j\in U} p(j)$. By convention we have $p(\emptyset) = 0$.

Given a feasible schedule S of jobs in N, we denote by $C_j(S)$ the completion time of job j in S. If no confusion arises, the reference to schedule S is omitted and we simply write C_j. We are interested in finding a feasible schedule S^* that minimizes the makespan, i.e., $C_{\max}(S^*) = \min_S \max\{C_j(S) : j \in N\}$.

The quality of an approximation algorithm H, which delivers a feasible schedule S^H, is measured by bounding the ratio $C_{\max}(S^H)/C_{\max}(S^*)$. A polynomial-time algorithm is said to be a ϱ-approximation algorithm if the following inequality holds for all instances of the problem:

$$\frac{C_{\max}(S^H)}{C_{\max}(S^*)} \le \varrho.$$

A *polynomial-time approximation scheme* (PTAS) is a family of $(1+\varepsilon)$-approximation algorithms for any $\varepsilon > 0$ with running time bounded by a polynomial in the length of the problem input. If the running-time bound of a PTAS is also polynomial with respect to $1/\varepsilon$, then it is called a *fully polynomial-time approximation scheme* (FPTAS).

Approximability issues of problem $P|\mathrm{res}\cdot 111|C_{\max}$ are addressed in [13], where a so-called group technology approach is analyzed. According to this approach the jobs that require the same resource are grouped into a composite job. If the search for an approximate solution is restricted to the group technology schedules, i.e., schedules in which each composite job is not split and is processed as a block, it is shown in [13] how to find a group technology schedule S^{GP} such that the following inequality holds:

$$\frac{C_{\max}(S^{\mathrm{GP}})}{C_{\max}(S^*)} \le \frac{2m}{m+1}.$$

Moreover, the bound $\varrho = 2m/(m+1)$ is the best possible for the group technology schedules. For problem $P2|\mathrm{res}\cdot 111|C_{\max}$, an improved algorithm presented in [13] behaves as a $(6/5)$-approximation algorithm. The algorithm allows a composite job to be split at most once. Another algorithm with the worst-case bound $\varrho = 2m/(m+1)$ is also given in [6], although it apparently requires more than $O(n \log n)$, the running time for the algorithm in [13].

In developing our FPTAS for $P2|\mathrm{res}\cdot 111|C_{\max}$, we will make a novel use of existing research results on open shop scheduling. In such a problem, we are given a set $N = \{1, 2, \ldots, n\}$ of jobs and each job $j \in N$ consists of $m \ge 2$ operations $\{O_{1j}, \ldots, O_{mj}\}$, where operation O_{ij} needs to be processed on machines M_i for an (uninterrupted) $p_{ij} \ge 0$ time units, $i = 1, \ldots, m$. A feasible schedule is an assignment of all operations to their designated machines in such a way that at most one operation of each job can be processed at a time and a machine can process no more than one operation at a time. A job is completed if and only if all its operations are completed. The makespan of an open shop schedule is defined as the maximum of all job completion times.

A comprehensive review on the complexity and approximation for open shop scheduling problem $O||C_{\max}$ and its variations can be found in [14]. For our purposes, in this note we present some facts on problem $O2||C_{\max}$ with the machine number fixed at $m = 2$. For a job $j \in N$, denote $a(j) = p_{1j}$ and $b(j) = p_{2j}$. Similarly, for any subset $U \subseteq N$ we denote $a(U) = \sum_{j\in U} a(j)$. The same applies to $b(U)$. Since the makespan cannot be less than either the total of the processing times of all jobs on each machine or the completion time of the last operation of each job, the following lower bound holds on the makespan of any schedule S^O:

$$C_{\max}(S_O) \ge \max\{\max\{a(j) + b(j) : j \in N\},\ a(N),\ b(N)\}.$$

Problem $O2||C_{\max}$ is solvable in $O(n)$ time. Five different algorithms based on different ideas are known. Listed in the chronological order, these algorithms are due to Gonzalez and Sahni [5], Pinedo and Schrage [11], de Werra [3], Soper [12] and Khramova and Chernych [10].

Our FPTAS for problem $P2|\mathrm{res}\cdot 111|C_{\max}$ will start with an FPTAS for the subset-sum problem, which is stated as the following special case of the knapsack problem: Given a set of positive numbers $\{w_j : j \in N'\}$ and a capacity c, find a subset of these numbers with their total value maximized without exceeding capacity c. The problem is formulated as follows:

$$\begin{array}{ll} \max & \sum_{j\in N'} w_j x_j \\ \text{s.t.} & \sum_{j\in N'} w_j x_j \le c, \\ & x_j \in \{0,1\},\ j \in N'. \end{array} \tag{1}$$

Theorem 1 *([8,9]). The subset-sum problem (1) admits an FPTAS such that, for any given $\varepsilon > 0$, it finds one of the following two solutions in $O(n/\varepsilon)$ time: (a) an optimal solution $\mathbf{x}^* = (x_j^* : j \in N')$ such that*

$$\sum_{j\in N'} w_j x_j^* < (1-\varepsilon)c,$$

or (b) an approximate solution $\mathbf{x}^\varepsilon = (x_j^\varepsilon : j \in N')$ such that

$$(1-\varepsilon)c \le \sum_{j\in N'} w_j x_j^\varepsilon \le c. \tag{2}$$

3. Approximation scheme

The FPTAS we are to present for problem $P2|\mathrm{res}\cdot 111|C_{\max}$ consists of two phases. Given any instance $\mathcal{I}$ of our problem, the first phase applies a known FPTAS for the subset-sum problem to transform $\mathcal{I}$ to an instance $\mathcal{I}'$ of problem $O2||C_{\max}$. The two operations of each job in instance $\mathcal{I}'$ is a split of the composite of all those jobs that require a same resource. This way a feasible schedule for $\mathcal{I}'$ will guarantee that the two split parts of the composite job will not be processed on both machines at any point of time, which translates to that no two jobs in $\mathcal{I}$ that require a same resource will be processed simultaneously.

The second phase of our FPTAS uses an algorithm for problem $O2||C_{\max}$ to find an optimal (open shop) schedule S_O for instance $\mathcal{I}'$. We then transform S_O to a near optimal schedule S_R for $\mathcal{I}$ of our original problem.

3.1. The trivial cases

Given any instance of problem $P2|\mathrm{res}\cdot 111|C_{\max}$, let $L = p(N)/2$ be the average machine load and let S_R^* be an optimal schedule. Note that no two jobs that require the same resource can be processed in parallel on both machines. On the other hand, the average machine load L and the largest processing time of jobs that require no resource are apparently lower bounds for the makespan of schedule S_R^*, we have

$$C_{\max}(S_R^*) \ge \max\left\{L,\ \max\left\{p(j) : j \in N^0\right\},\ \max\{p(N^k) : 1 \le k \le q\}\right\}. \tag{3}$$

If there exists a job $\ell \in N^0$ such that $p(\ell) > L$, then the following schedule is optimal: job ℓ is processed on one of the machines and the remaining jobs are processed on the other machine. Similarly, if for some k, $1 \le k \le q$, the total processing time $p(N^k)$ of the jobs that require resource k exceeds L, then an optimal schedule is to process the jobs of set N^k by one machine and process the remaining jobs by the other machine. Thus, in what follows, we exclude these two trivial cases and assume that

$$L \geq \max\left\{\max\left\{p(j) : j \in N^0\right\}, \max\{p(N^k) : 1 \leq k \leq q\}\right\}. \quad (4)$$

3.2. General case

Under the assumption (4), we present a formal description of our FPTAS below. Given instance $\mathcal{I}$ of problem $P2|\text{res} \cdot 111|C_{\max}$, for any $\varepsilon > 0$ we describe how to find a schedule S_R^ε such that

$$C_{\max}\left(S_R^\varepsilon\right) \leq (1+\varepsilon)\, C_{\max}\left(S_R^*\right). \quad (5)$$

Algorithm FPTAS

Step 1. Given any instance $\mathcal{I}$ of problem $P2|\text{res} \cdot 111|C_{\max}$ that satisfies (4), consider the subset-sum problem (1) with $N' = N$, $w_j = p(j)$ for any $j \in N$, and $c = L$. Use Theorem 1 and apply an FPTAS to find a solution $\tilde{\mathbf{x}} = (\tilde{x}_j : j \in N)$. Let C^ε be the value of the objective function. Let $H_1 = \{j \in N : \tilde{x}_j = 1\}$ and $H_2 = \{j \in N : \tilde{x}_j = 0\}$.

Step 2. Renumber the jobs in N if needed so that $N^0 = \{1, \dots, n_0\}$, where $n_0 = |N^0|$. Let $T = \{1, \dots, t\}$ with $t = q + n_0$. First, construct an instance $\mathcal{I}'$ of the auxiliary problem $O2||C_{\max}$ as follows: The job set of instance $\mathcal{I}'$ is $\{V_k : k \in T\}$, where the two operations of job V_k $(1 \leq k \leq q)$ have respective processing times of

$$a(V_k) = \textstyle\sum_{j \in N^k \cap H_1} p(j), \text{ and } b(V_k) = \textstyle\sum_{j \in N^k \cap H_2} p(j).$$

For any $j \in N^0$, the two operations of job V_{q+j} have the following processing times: $a(V_{q+j}) = p(j)$ and $b(j) = 0$ if $j \in N^0 \cap H_1$; $a(V_{q+j}) = 0$ and $b(j) = p(j)$ if $j \in N^0 \cap H_2$. Then use any suitable linear-time algorithm (as given in [5,11,3,12,10]) to find an optimal (open shop) schedule S_O^* for instance $\mathcal{I}'$.

Step 3. Transform S_O^* into a feasible schedule S_R^ε for the original problem $P2|\text{res} \cdot 111|C_{\max}$ as follows: For $k = 1, \dots, q$, replace the operation of each job V_k on machine M_1 by the block of jobs in $N^k \cap H_1$ and the operation of the same job on machine M_2 by the block of jobs in $N^k \cap H_2$. Then replace each job V_{q+j} by job $j \in N^0$.

Theorem 2. *Algorithm FPTAS creates a (feasible) schedule S_R^ε for problem $P2|\text{res} \cdot 111|C_{\max}$ such that inequality (5) holds. The running time of the algorithm is $O(n/\varepsilon)$.*

Proof. At the end of Step 1, the value C^ε and sets H_1 and H_2 are determined. Consider problem $O2||C_{\max}$ of scheduling the composite jobs defined in Step 2.

According to our construction of the auxiliary problem $O2||C_{\max}$, we have

$$\sum_{k \in T} a(V_k) = C^\varepsilon \leq L \leq \sum_{k \in T} b(V_k) = 2L - C^\varepsilon,$$

where the first and second inequalities are due to the constraint of the subset-sum problem together with the fact that

$$\sum_{k \in T} (a(V_k) + b(V_k)) = p(N) = 2L.$$

Consequently, from (4) and the fact that $p(N^k) = a(V_k) + b(V_k)$ we obtain

$$2L - C^\varepsilon = \max\left\{\sum_{k \in T} a(V_k), \sum_{k \in T} b(V_k)\right\} \geq L \geq \max\{a(V_k) + b(V_k) : k \in T\}.$$

In other words, the optimal open shop schedule S_O^* obtained at the end of Step 2 has a makespan equal to the larger machine load, i.e.,

$$C_{\max}\left(S_O^*\right) = 2L - C^\varepsilon.$$

Notice that the feasibility of schedule S_O^* guarantees that if (composite) job V_k consists of two operations of strictly positive duration each, the two operations are not executed at the same time, which implies that after transforming schedule S_O^* to schedule S_R^ε of the original problem $P2|\text{res} \cdot 111|C_{\max}$, no two jobs of N^k will be processed at the same time on both machines for any $k = 1, \dots, q$. In other words, S_R^ε is feasible and

$$C_{\max}\left(S_R^\varepsilon\right) = 2L - C^\varepsilon. \quad (6)$$

Now let us look closely at the solution $\tilde{\mathbf{x}} = (\tilde{x}_j : j \in N)$ that is found at Step 1. There are two cases (a) and (b) when Theorem 1 is applied. In the case of (a), solution $\tilde{\mathbf{x}}$ is optimal for problem (1). In other words, in splitting the total workload of $p(N)$ between the two identical machines, the split of (H_1, H_2) minimizes the larger machine load to $2L - C^\varepsilon \geq L \geq C^\varepsilon$, which together with (6) implies that schedule S_R^ε is optimal for problem $P2|\text{res} \cdot 111|C_{\max}$.

Now let us consider case (b) when Theorem 1 is applied. In this case, according to (2), we have $C^\varepsilon \geq (1-\varepsilon)\, L$, which together with (6) implies that

$$C_{\max}\left(S_R^\varepsilon\right) = 2L - C^\varepsilon \leq (1+\varepsilon)\, L \leq (1+\varepsilon)\, C_{\max}\left(S_R^*\right),$$

where the last inequality comes from (3). Hence (5) holds. □

4. Concluding remarks

The fully polynomial-time approximation scheme we have developed in this note for problem $P2|\text{res} \cdot 111|C_{\max}$ provides a stepping stone for us to effectively approximate the general problem $Pm|\text{res}\, \rho_1\rho_2\rho_3\rho_4|C_{\max}$ for any fixed $m \geq 2$ or even problem $P|\text{res}\, \rho_1\rho_2\rho_3\rho_4|C_{\max}$ for variable m. A natural next challenge is to determine whether problem $Pm|\text{res} \cdot 111|C_{\max}$ admits either an FPTAS or a PTAS.

Acknowledgement

This piece of research was initiated when the second author visited Warwick Business School. Support of this visit provided by the School (IBROPBC1) is highly appreciated.

References

[1] J. Błażewicz, J.K. Lenstra, A.H.G. Rinnooy Kan, Scheduling subject to resource constraints: classification and complexity, Discrete Appl. Math. 5 (1983) 11–24.
[2] B. Chen, C.N. Potts, G.J. Woeginger, A review of machine scheduling: complexity, algorithms and approximability, in: D.-Z. Du, P. Pardalos (Eds.), Handbook of Combinatorial Optimization, Volume 3, Kluwer Academic Publishers, 1998, pp. 21–169.
[3] D. de Werra, Graph-theoretical models for preemptive scheduling, in: R. Slowinski, J. Weglarz (Eds.), Advances in Project Scheduling, Elsevier, Amsterdam, 1989, pp. 171–185.
[4] E.B. Edis, C. Oguz, I. Ozkarahan, Parallel machine scheduling with additional resources: notation, classification, models and solution methods, Eur. J. Oper. Res. 230 (2013) 449–463.
[5] T. Gonzalez, S. Sahni, Open shop scheduling to minimize finish time, J. Assoc. Comput. Mach. 23 (1976) 665–679.

[6] E. Hebrard, M.-J. Hugueta, N. Jozefowiez, A. Maillard, C. Pralet, G. Verfaillie, Approximation of the parallel machine scheduling problem with additional unit resources, Discrete Appl. Math. 215 (2016) 126–135.
[7] T. Janssen, C. Swennenhuis, A. Bitar, T. Bosman, D. Gijswijt, L. van Iersel, S. Dausère-Pérèz, C. Yugma, Parallel machine scheduling with a single resource per job, arXiv:1809.05009v3, 2018.
[8] H. Kellerer, R. Mansini, U. Pferschy, M.G. Speranza, An efficient fully polynomial approximation scheme for the Subset-Sum problem, J. Comput. Syst. Sci. 66 (2003) 349–370.
[9] H. Kellerer, U. Pferschy, D. Pisinger, Knapsack Problems, Springer, Berlin, 2004.
[10] A.P. Khramova, I. Chernykh, A new algorithm for the two-machine open shop and the polynomial solvability of a scheduling problem with routing, J. Sched. 24 (2021) 405–412.
[11] M. Pinedo, L. Schrage, Stochastic shop scheduling: a survey, in: M.A.H. Dempster, et al. (Eds.), Deterministic and Stochastic Scheduling, Riedel, Dordrecht, 1982, pp. 181–196.
[12] A.J. Soper, A cyclical search for the two machine flow shop and open shop to minimise finishing time, J. Sched. 18 (2015) 311–314.
[13] V.A. Strusevich, Approximation algorithms for makespan minimization on identical parallel machines under resource constraints, J. Oper. Res. Soc. 72 (2021) 2135–2146.
[14] V.A. Strusevich, Complexity and approximation of open shop scheduling to minimize the makespan: a review of models and approaches, Comput. Oper. Res. 144 (2022) 105732.