Operations Research Letters 44 (2016) 354–358

Contents lists available at ScienceDirect

Operations Research Letters

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

Complexity of interval minmax regret scheduling on parallel identical machines with total completion time criterion

Maciej Drwal [a,*], Roman Rischke [b]

[a] Department of Computer Science, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

[b] Center for Mathematics, Technische Universität München, Boltzmannstr. 3, 85747 Garching bei München, Germany

ARTICLE INFO

Article history:
Received 13 December 2014
Received in revised form
9 December 2015
Accepted 8 March 2016
Available online 16 March 2016

Keywords:
Robust optimization
Scheduling
Uncertainty
Computational complexity

ABSTRACT

We consider the problem of scheduling jobs on parallel identical machines, where only interval bounds of processing times of jobs are known. The optimality criterion of a schedule is the total completion time. In order to cope with the uncertainty, we consider the maximum regret objective and seek a schedule that performs well under all possible instantiations of processing times. We show how to compute the maximum regret, and prove that its minimization is strongly NP-hard.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Robust optimization has been applied to many combinatorial optimization problems, since in practical applications input data to most problems can be rarely given precisely. This is true in the context of scheduling, as in many actual execution environments (e.g., computer systems, transportation, manufacturing) processing times of tasks are not known exactly, but their values can fluctuate within certain bounds. Moreover, very often no good assumptions can be made even regarding their probability distributions. In such circumstances we would like to find a solution that is the best in the worst possible scenario of events. Such solutions can be characterized in terms of the maximum regret criterion [7,2,5]. Solutions that minimize the maximum regret are often much more reliable than the ones obtained by ignoring parameter uncertainty. However, in many cases, finding a robust solution for uncertain data is more difficult and may require more computational resources.

We apply the minmax regret approach to the problem of scheduling on parallel identical machines to minimize the total completion time (sum of the completion times of all jobs) with interval uncertainty in the job processing times. The problem under consideration is denoted INTERVAL MINMAX REGRET $P \parallel \sum C_i$, using the notation from standard scheduling theory. The deterministic version of this problem can be solved in polynomial time by applying the *shortest processing time first* rule [10]. However, its minmax regret version becomes NP-hard even for a single machine, i.e., INTERVAL MINMAX REGRET $1 \parallel \sum C_i$. In [8] it is shown that even when the midpoints of all intervals are equal and the number of jobs is odd, finding an optimal robust sequence on a single machine is weakly NP-hard. Surprisingly, for an even number of jobs this problem is polynomially solvable. Thus the case in which the number of machines is given as part of the input can be no easier. Recently, Conde [3] indicated a simple reduction from the minmax regret assignment problem [1] of m jobs to m machines, which implies that in case of m parallel unrelated machines (INTERVAL MINMAX REGRET $R \parallel \sum C_i$) the problem is strongly NP-hard. However, this reduction does not suffice to prove hardness for the case of parallel identical machines. In this paper, we extend the aforementioned complexity results by showing that the problem is strongly NP-hard even on identical parallel machines.

2. Problem formulation

In the scheduling problem $P \parallel \sum C_i$ we are given m identical parallel machines (processors) for processing n jobs, where each job i has an integer processing time p_i, $i = 1, \ldots, n$. If not stated otherwise, we assume that all $p_i \geq 0$. Each job has to be assigned to exactly one machine. Let π_j denote a vector of integers, where $\pi_j(k)$ is the index of the job scheduled on the jth machine as the kth to the last (jobs on each machine are scheduled starting from

* Corresponding author.
E-mail addresses: maciej.drwal@pwr.edu.pl (M. Drwal), rischke@ma.tum.de (R. Rischke).

http://dx.doi.org/10.1016/j.orl.2016.03.005
0167-6377/© 2016 Elsevier B.V. All rights reserved.

time zero and without idle times). Let n_j denote the number of elements in π_j, i.e., the number of jobs assigned to machine j. The *completion time* of the job scheduled as kth to the last on machine j is $C_{j,k} = \sum_{i=k}^{n_j} p_{\pi_j(i)}$ ($C_{j,k} = 0$ if there is no such job). The objective is to minimize the total sum of completion times (also called the *total flow time*), expressed as:

$$F(\pi) = \sum_{j=1}^{m}\sum_{k=1}^{n_j} C_{j,k} = \sum_{j=1}^{m}\sum_{k=1}^{n_j} kp_{\pi_j(k)}, \tag{1}$$

where $\pi = [\pi_1, \dots, \pi_m]$ is called a *schedule*. We will sometimes refer to this problem formulation as the *deterministic version* of the scheduling problem.

The definition of the minmax regret version of this problem with interval uncertainty, denoted as INTERVAL MINMAX REGRET $P \parallel \sum C_i$, differs as follows. Instead of having exact processing times p_i, we are now given only intervals $[p_i^-, p_i^+]$, $i = 1, \dots, n$, to which the actual processing times belong. We denote by $S = [p_1^S, \dots, p_n^S]$ any vector that satisfies $p_i^- \le p_i^S \le p_i^+$ for all $i = 1, \dots, n$. Such a vector will be called a *scenario*. For any schedule π and scenario S we define the value of *regret* as $Z(\pi, S) := F(\pi, S) - F^*(S)$, where $F(\pi, S)$ is the objective function (1) from the deterministic version of the problem $P \parallel \sum C_i$ with input data S, and $F^*(S)$ is the value of an optimal solution of this problem. The objective of INTERVAL MINMAX REGRET $P \parallel \sum C_i$ is to minimize over schedules the maximum of regret over scenarios:

$$Z^* = \min_{\pi}\max_{S}\left(F(\pi, S) - F^*(S)\right). \tag{2}$$

The above minmax regret formulation is a *robust optimization* formulation of the scheduling problem. A schedule minimizing the maximum regret will be called *robust optimal*.

3. Computation of maximum regret

The deterministic version of the considered problem is solvable in polynomial time (see [10, Theorem 5.3.1]). An optimal schedule can be obtained by first sorting all n jobs in order of non-decreasing processing times, and then we assign the first unassigned job in the list to the least loaded machine, i.e., to the machine with the smallest current makespan. Repeating this procedure until all jobs are assigned gives the desired schedule.

We show that for a fixed schedule π it is possible to compute the value of maximum regret $Z(\pi) = \max_S Z(\pi, S)$ in polynomial time. The method is analogous to the one presented in [3] for parallel unrelated machines, with the main difference that in the identical machines case the input data contains a single interval $[p_i^-, p_i^+]$ instead of m intervals given in the case of unrelated machines. Thus we omit the simple proof of correctness of Formulas (3)–(5).

Let us encode a feasible solution π of the considered problem in terms of binary variable $\mathbf{x}$ as follows: let $x_{ijk} = 1$ iff the ith job is processed on the jth machine as the kth to the last.

For any feasible schedule $\mathbf{x}$ the maximum regret can be computed as:

$$\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{k=1}^{n} kp_i^+ x_{ijk} - \min_{\mathbf{y}} \sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{k=1}^{n} c_{ijk}(\mathbf{x}) y_{ijk}, \tag{3}$$

where

$$c_{ijk}(\mathbf{x}) = kp_i^- + (p_i^+ - p_i^-)\sum_{l=1}^{m}\sum_{r=1}^{n} \min\{r, k\} x_{ilr}. \tag{4}$$

The minimization in (3) is equivalent to the minimum assignment problem and thus can be solved in polynomial time, using e.g. the Hungarian method [9]. Variable $\mathbf{y}$ is an $n \times (mn)$ permutation matrix.

Let $x_{ij} = k$ iff $x_{ijk} = 1$, and $y_{ij} = k$ iff $y_{ijk} = 1$. For a fixed $\mathbf{x}$, given a solution $\mathbf{y}$ of the minimization in (3), the worst-case scenario can be obtained as:

$$p_i = \begin{cases} p_i^+ & \text{if } \sum_{j=1}^{m}(x_{ij} - y_{ij}) \ge 0, \\ p_i^- & \text{otherwise.} \end{cases} \tag{5}$$

4. Properties of optimal solutions

Denote by $Z(\pi)$ the solution value of the INTERVAL MINMAX REGRET $P \parallel \sum C_i$ problem for a schedule $\pi = [\pi_1, \dots, \pi_m]$. The maximum regret can be written as:

$$\begin{aligned} Z(\pi) &= \max_S \left(F(\pi, S) - F^*(S)\right) \\ &= \max_S \sum_{j=1}^{m}\left(F_j(\pi_j, S) - F_j^*(S)\right) = \sum_{j=1}^{m} Z_j(\pi_j). \end{aligned}$$

Here $F_j(\pi_j, S) = \sum_{k=1}^{n_j} kp^S_{\pi_j(k)}$ is the sum of completion times of the jobs on machine j under scenario S, $n_j = |\pi_j|$ is the number of jobs assigned to machine j, $F_j^*(S)$ is the sum of completion times of the jobs on machine j in an optimal solution under scenario S, and finally, $Z_j(\pi_j)$ is the difference between $F_j(\pi_j, S)$ and $F_j^*(S)$ for a scenario S that maximizes the total regret. Let π^* denote an optimal robust solution, i.e., a schedule that minimizes Z.

Consider a worst-case scenario for any schedule π (see Eq. (5)). This scenario defines an instance of the deterministic version of the problem. An optimal solution of this deterministic problem is called a *worst-case alternative* for π.

From now on, we consider only instances satisfying the following assumption.

Assumption 1. The number of jobs n is divisible by the number of machines m, i.e., there exists an integer $n_0 > 0$ such that $n = m \cdot n_0$.

In particular, if $m|n$, then any schedule has a worst-case alternative with an equal number of jobs on each machine. Moreover, the following is true.

Lemma 1. *If $m|n$, then in an optimal robust schedule every machine is assigned the same number of jobs.*

Proof. Let π_2 be any schedule with different number of jobs on at least two machines. Under a fixed scenario S, there exists a schedule π_1 with an equal number of jobs on each machine, such that $F(\pi_1, S) < F(\pi_2, S)$. This follows from the fact that we can construct π_1 from π_2 by performing a sequence of the following job displacements: from the machine with the longest schedule remove the job from the first position and insert it at the first position on the machine with the least number of jobs (the multipliers k in (1) of the remaining jobs are unchanged, but the multiplier of the moved job may decrease; the last such displacement must be performed between two machines that differ in the number of jobs by 2, thus the multiplier of that job decreases by 1, and the overall cost of the schedule decreases).

Let us denote by S^{π} the worst-case scenario for π. Then we get:

$$\begin{aligned} Z(\pi_1) &= F(\pi_1, S^{\pi_1}) - F^*(S^{\pi_1}) < F(\pi_2, S^{\pi_1}) - F^*(S^{\pi_1}) \\ &\le F(\pi_2, S^{\pi_2}) - F^*(S^{\pi_2}) = Z(\pi_2). \end{aligned}$$

The last inequality follows from the fact that S^{π_1} is not necessarily the worst-case scenario for π_2, thus by definition the value of regret $Z(\pi_2, S^{\pi_1})$ is no greater than that of the maximum

Download English Version:

https://daneshyari.com/en/article/1142140

Download Persian Version:

https://daneshyari.com/article/1142140

Daneshyari.com