European Journal of Operational Research 240 (2015) 355–360

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Scheduling: Agreement graph vs resource constraints

Mohamed Bendraouche [a,b,*], Mourad Boudhar [b], Ammar Oulamara [c]

[a] Faculty of Sciences, Saad Dahleb University, Route de Soumaa, BP 270, Blida, Algeria
[b] RECITS Laboratory, USTHB University, BP 32 El-Alia, Bab-Ezzouar, Algiers, Algeria
[c] Loria Laboratory, University of Lorraine, Nancy, France

ARTICLE INFO

Article history:
Received 5 March 2013
Accepted 1 July 2014
Available online 9 July 2014

Keywords:
Scheduling
Complexity theory
Identical machines
Agreement graph
Resource constraints

ABSTRACT

We investigate two scheduling problems. The first is scheduling with agreements (SWA) that consists in scheduling a set of jobs non-preemptively on identical machines in a minimum time, subject to constraints that only some specific jobs can be scheduled concurrently. These constraints are represented by an agreement graph. We extend the NP-hardness of SWA with three distinct values of processing times to only two values and this definitely closes the complexity status of SWA on two machines with two fixed processing times. The second problem is the so-called resource-constrained scheduling. We prove that SWA is polynomially equivalent to a special case of the resource-constrained scheduling and deduce new complexity results of the latter.

1. Introduction

In this paper, two scheduling problems are addressed. The first is scheduling with agreements (Bendraouche & Boudhar, 2012), SWA in short, whose input consists of a set $V = \{J_1, J_2, \ldots, J_n\}$ of n jobs where each job J_i $(i = 1, \ldots, n)$ has a processing time p_i and a release date r_i. These jobs must be scheduled on a set of m identical machines. We assume that there is a graph $G = (V, E)$ over the jobs, called the agreement graph. Each edge in E models a pair of agreeing jobs that can be scheduled concurrently on different machines. The agreement constraints are such that only agreeing jobs can be scheduled concurrently. A schedule is an assignment of jobs to the machines which specifies for each job the time interval and the machine on which this job is to be processed. A feasible schedule is a non-preemptive one which respects the agreement constraints. The aim is to find a feasible schedule that minimizes the makespan. Motivations and applications of this problem can be found in Bendraouche and Boudhar (2012), Baker and Coffman (1996), Halldorsson et al. (2003), Bodlaender and Jansen (1995), Gardi (2009) and Even, Halldorson, Kaplan, and Ron (2009).

To be in concordance with the scheduling notation, the SWA problem is also denoted $P|AgreeG = (V, E), \ldots|C_{max}$.

The second problem is (discrete) resource-constrained scheduling defined by a set $J = \{J_1, \ldots, J_n\}$ of n jobs, having processing times $p_1, \ldots, p_n$ and release dates $r_1, \ldots, r_n$ respectively. All jobs must be scheduled non-preemptively on m identical machines. Besides the machines, we suppose there are s types of additional renewable resources $R_1, R_2, \ldots, R_s$, which are available in $u_1, u_2, \ldots, u_s$ units respectively. Each job requires for its processing specific amounts of resources. Thus, each job J_i $(i = 1, \ldots, n)$ is characterized by (1) the resource requirement vector $R(J_i) = [R_1(J_i), \ldots, R_s(J_i)]$ where $0 \leqslant R_k(J_i) \leqslant u_k$ $(k = 1, \ldots, s)$ represents the number of units of resource R_k required for the processing of the job J_i, and (2) the processing time p_i and its release date r_i. Moreover, we assume that all required resources are granted to a job before its processing begins and they are returned by the job after its completion. In the literature this problem is denoted by $P|res\lambda\delta\rho|C_{max}$ where $res\lambda\delta\rho$ describes the additional resources as introduced in Blazewicz, Lenstra, and Rinnooy Kan (1983). The entry $\lambda \in \{\cdot, s\}$ indicates the number of different resources, $\delta \in \{\cdot, o\}$ specifies the resources availabilities and $\rho \in \{\cdot, r\}$ indicates the resource requirements. If $\lambda = s, \delta = o, \rho = r$ this means that the number of resources is bounded by s, the resource availabilities is bounded by o and the resource requirements is bounded by r respectively. If an entry equals "·", the corresponding amount is specified by the input.

Another related problem worth to be cited is the so-called multiprocessor jobs scheduling with model "fix". In this problem n jobs are to be processed on m dedicated processors $M_1, M_2, \ldots, M_m$. Each job J_i requires during its processing time period p_i a subset

* Corresponding author at: Faculty of Sciences, Saad Dahleb University, Route de Soumaa, BP 270, Blida, Algeria.

E-mail addresses: mbendraouche@yahoo.fr (M. Bendraouche), mboudhar@yahoo.fr (M. Boudhar), Ammar.Oulamara@loria.fr (A. Oulamara).

http://dx.doi.org/10.1016/j.ejor.2014.07.003
0377-2217/

$\mu_i \subseteq \{M_1, M_2, \ldots, M_m\}$ of processors simultaneously. Following the notation of Blazewicz, Ecker, Pesch, Schmidt, and Weglarz (2001), this problem is denoted $P|fix_j|C_{max}$. This multiprocessor jobs scheduling can be seen as a special case of SWA problem with agreement graph $G = (V, E)$ such that V is the set formed by the n jobs and that $(J_i, J_j) \in E \iff \mu_i \cap \mu_j = \phi$. Furthermore, the number of machines in the SWA problem must be sufficiently large e.g. $\min\{m, n\}$.

The remainder of this paper is organized as follows. Section 2 describes the literature review. Section 3, presents new complexity results for the SWA problem in the case of two machines with at most two distinct values of processing times. In Section 4, the polynomial equivalence between SWA and resource-constrained problem is proved from which new complexity results of the latter are derived. Concluding remarks constitute Section 5.

2. Literature review

The SWA problem with two machines and arbitrary processing times is NP-hard even if the agreement graph is complete (Garey & Johnson, 1979).

In the case of unit processing times, the SWA problem is equivalent to finding a partition of a given graph into a minimum number of cliques, each with size at most m. This problem is equivalent to the Mutual Exclusion Scheduling problem introduced by Baker and Coffman (1996). The SWA problem with two machines and unit processing times is polynomial since a maximum matching in the agreement graph yields an optimal schedule (see Baker & Coffman, 1996).

Even et al. (2009) have considered the SWA problem under the name Scheduling With Conflicts (SWC in short), by considering the complement of the agreement graph called the conflict graph. Thus, these two problems are equivalent.

Table 1 summarizes the other recent complexity results of the SWA problem where $AgreeG = (V, E)$ and $AgreeG = (S_1, S_2; E)$ denote an arbitrary agreement graph and arbitrary bipartite agreement graph respectively.

Resource-constrained scheduling problems are largely considered in the literature. An overview and complexity classification of scheduling problems with additional renewable resources can be found in Blazewicz et al. (1983). Table 2 summarizes all the recent complexity results concerning this problem. Whereas, Table 3 reports some of the recent complexity results concerning multiprocessor scheduling with model "fix".

3. Scheduling With Agreements (SWA) on two machines with two distinct processing times

In Bendraouche and Boudhar (2012), the authors proved that the SWA problem with two machines and $p_i \in \{1, 2, 3\}$ is strongly NP-hard even for arbitrary bipartite agreement graphs. In this section, we show that this result remains true even with two distinct values of processing times, and this will definitely close the complexity status for the SWA problem in the case of two machines.

Table 1
Previous complexity results of the SWA problem.

Problem	Complexity	Reference
$P2\|AgreeG = (V, E),$ $p_i \in \{1, 2\}\|C_{max}$	Poly.	Even et al. (2009)
$P2\|AgreeG = (S_1, S_2; E),$ $p_i \in \{1, 2, 3\}\|C_{max}$	Strongly NP-hard	Bendraouche and Boudhar (2012)
$P2\|AgreeG = (S_1, S_2; E),$ $r_i \in \{0, r\}, p_i \in \{1, 2\}\|C_{max}$	Strongly NP-hard	Bendraouche and Boudhar (2012)
$P2\|AgreeG = (S_1, S_2; E),$ $p_{S_1} = 1\|C_{max}$	Poly.	Bendraouche and Boudhar (2012)

Table 2
Previous complexity results of the resource-constrained scheduling.

Problem	Complexity	Reference
$P2\|res\,1\cdot\cdot, p_i = 1\|C_{\max}$	$O(nlogn)$	Blazewicz et al. (2001)
$P2\|res\,1\cdot\cdot, r_i, p_i = 1\|C_{\max}$	sNP-hard	Blazewicz et al. (1986)
$P2\|res\cdot\cdot\cdot, p_i = 1\|C_{\max}$	$O(n^{2.5})$	Garey and Johnson (1975)
$P2\|res\cdot 11, r_i, p_i = 1\|C_{\max}$	sNP-hard	Blazewicz et al. (1986)
$P3\|res\,1\cdot\cdot, p_i = 1\|C_{\max}$	sNP-hard	Garey and Johnson (1975)
$P3\|res\cdot 11, r_i, p_i = 1\|C_{\max}$	sNP-hard	Blazewicz et al. (1983)
$P\|res\,1\cdot 1, r_i, p_i = 1\|C_{\max}$	$O(n)$	Blazewicz (1978)
$P\|res\,spr, p_i = 1\|C_{\max}$	$O(n)$	Blazewicz and Ecker (1983)

Table 3
Some previous results of multiprocessor jobs model "fix" scheduling.

Problem	Complexity	Reference
$P\|fix_j\|C_{\max}$ and $\|fix_j\| = 2$	20 NP-h cases 23 poly. cases	Coffman et al. (1985)
$P\|fix_j\|C_{\max}$ and $\|fix_j\| \in \{1, 2\}$	9 NP-h cases 9 poly. cases	Kubale (1987)
$P\|fix_j, p_j = 1\|C_{\max}$	sNP-hard	Krawczyk and Kubale (1985)
$P3\|fix_j\|C_{\max}$	sNP-hard	Blazewicz et al. (1994) Hoogeveen and Van de Velde (1994)
$P2\|fix_j, p_j = 1\|C_{\max}$ $P3\|fix_j, p_j = 1\|C_{\max}$ $P4\|fix_j, p_j = 1\|C_{\max}$	$O(n)$	Bianco et al. (1994)
$P5\|fix_j, p_j = 1\|C_{\max}$	$O(n^{2.5})$	Bianco et al. (1994)
$P\|fix_j, p_j = 1\|C_{\max}$	sNP-hard	Hoogeveen and Van de Velde (1994)

We shall denote the general problem, with two distinct processing times by $P2|AgreeG = (V, E), p_i \in \{a, 2a + b\}|C_{max}$. In the rest of this paper, we assume that $a \geqslant 1$ and $2a + b > a$, which is equivalent to $b > -a$ and will consider all cases: $b = 0, b \geqslant 1$ and $-a < b < 0$. When $b = 0$, this corresponds to $p_i \in \{a, 2a\}$. In Even et al. (2009), the authors proved that the problem with $p_i \in \{1, 2\}$ is polynomial by using the maximum matching technique. By the same argument one can easily verify that this result is also true when $p_i \in \{a, 2a\}$, and this simply can be done by replacing the length of unit jobs by a. Thus, in the following we shall concentrate on the other two cases: $b \geqslant 1$ and $-a < b < 0$.

3.1. Case $p_i \in \{a, 2a + b\}$ and $b \geqslant 1$

In the next result, we prove that for two machines, $p_i \in \{a, 2a + b\}$ and $b \geqslant 1$ the SWA problem is strongly NP-hard even for arbitrary bipartite agreement graphs. The candidate used in the reduction process is the 3-Dimensional Matching Problem (3-DM), which is known to be NP-complete (see for e.g. Garey & Johnson, 1979). This problem is defined as follows.

Instance: a set $M \subseteq X \times Y \times Z$ where X, Y and Z are mutually disjoint sets having the same cardinality q.

Question: does M contain a subset $M' \subseteq M$ such that $|M'| = q$ and no two elements of M' agree in any coordinate?

Theorem 1. *The SWA problem $P2|AgreeG = (S_1, S_2; E), p_i \in \{a, 2a + b\}|C_{max}$ is NP-hard in the strong sense for any two distinct values a and $2a + b$ with $b \geqslant 1$.*

Proof. The proof will be divided into three parts depending on the values of b, namely $b < a, b > a$ and $b = a$.

Case 1: $b < a$.

We use a reduction similar to that in Bendraouche and Boudhar (2012) with the following changes. Given an arbitrary instance of 3-DM, we construct the corresponding scheduling instance as follows. Let us construct the agreement graph $G = (V, E)$.