ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Theoretical Computer Science 362 (2006) 273–281

Theoretical Computer Science

www.elsevier.com/locate/tcs

Multi-agent scheduling on a single machine to minimize total weighted number of tardy jobs

T.C.E. Cheng[a,*], C.T. Ng[a], J.J. Yuan[b]

[a] *Department of Logistics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, People's Republic of China*

[b] *Department of Mathematics, Zhengzhou University, Zhengzhou, Henan 450052, People's Republic of China*

Received 29 December 2005; received in revised form 25 June 2006; accepted 9 July 2006

Communicated by Ding-Zhu Du

Abstract

We consider the feasibility model of multi-agent scheduling on a single machine, where each agent's objective function is to minimize the total weighted number of tardy jobs. We show that the problem is strongly NP-complete in general. When the number of agents is fixed, we first show that the problem can be solved in pseudo-polynomial time for integral weights, and can be solved in polynomial time for unit weights; then we present a fully polynomial-time approximation scheme for the problem.

Keywords: Scheduling; Multi-agent deterministic sequencing

1. Introduction and problem formulation

The following single-machine multi-agent scheduling problem was introduced by Agnetis et al. [1] and Baker and Smith [2]. There are several agents, each with a set of jobs. The agents have to schedule their jobs on a common processing resource, i.e., a single machine, and each agent wishes to minimize an objective function that depends on the completion times of his own set of jobs. The problem is either to find a schedule that minimizes a combination of the agents' objective functions or to find a schedule that satisfies each agent's requirements for his own objective function.

Scheduling is in fact concerned with the allocation of limited resources over time. Scheduling situations involving multiple customers (agents) competing for a common processing resource arise naturally in many settings. For example, in industrial management, the multi-agent scheduling problem is formulated as a sequencing game, where the objective is to devise some mechanisms to encourage the agents to cooperate with a view to minimizing the overall cost (see, for example, [3,5]). In project scheduling, the problem is concerned with negotiation to resolve conflicts whenever the agents find their own schedules unacceptable [6]. In telecommunication services, the problem is to do with satisfying the service requirements of individual agents, who compete for the use of a commercial satellite to transfer voice, image and data files to their clients [10].

* Corresponding author. Tel.: +852 2766 5216; fax: +852 2364 5245.
E-mail address: LGTCheng@polyu.edu.hk (T.C.E. Cheng).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2006.07.011

In the following we define the single-machine multi-agent scheduling problem in terms of common scheduling terminology. We are given m families of jobs $\mathcal{J}^{(1)}, \mathcal{J}^{(2)}, \ldots, \mathcal{J}^{(m)}$, where, for each i with $1 \leqslant i \leqslant m$,

$$\mathcal{J}^{(i)} = \left\{ J_1^{(i)}, J_2^{(i)}, \ldots, J_{n_i}^{(i)} \right\}.$$

The jobs in $\mathcal{J}^{(i)}$ are called the ith agent's jobs. Each job $J_j^{(i)}$ has a positive integral processing time (length) $p_j^{(i)}$, a positive integral due date $d_j^{(i)}$, and a positive integral weight $w_j^{(i)}$. All the jobs have a zero release time. The jobs will be processed on a single machine starting at time zero without overlapping and idle time between them. A schedule is a sequence of the jobs that specifies the processing order of the jobs on the machine. Under a schedule σ, the completion time of job $J_j^{(i)}$ is denoted by $C_j^{(i)}(\sigma)$; job $J_j^{(i)}$ is called tardy if $C_j^{(i)}(\sigma) > d_j^{(i)}$, and early otherwise; $U_j^{(i)}(\sigma) = 1$ if $J_j^{(i)}$ is tardy, and zero otherwise. For each job $J_j^{(i)}$, let $f_j^{(i)}(\cdot)$ be a non-decreasing function of the completion time of job $J_j^{(i)}$ (such an objective function is called *regular* in the scheduling literature).

In general, the ith agent's objective function $F^{(i)}(\sigma)$ has either one of the following two forms:

$$\text{max-form} \quad F^{(i)}(\sigma) = \max_{1 \leqslant j \leqslant n_i} f_j^{(i)}(C_j^{(i)}(\sigma)),$$

$$\text{sum-form} \quad F^{(i)}(\sigma) = \sum_{1 \leqslant j \leqslant n_i} f_j^{(i)}(C_j^{(i)}(\sigma)).$$

Furthermore, the single-machine multi-agent scheduling problem includes the following two models:

- *Feasibility model*: $1||F^{(i)} \leqslant Q_i, 1 \leqslant i \leqslant m$. In this model, the goal is to find a feasible schedule σ that satisfies $F^{(i)}(\sigma) \leqslant Q_i, 1 \leqslant i \leqslant m$.
- *Minimality model*: $1||\sum_{1 \leqslant i \leqslant m} F^{(i)}$. In this model, the goal is to find a schedule σ that minimizes $\sum_{1 \leqslant i \leqslant m} F^{(i)}(\sigma)$.

In this paper we always assume that $f_j^{(i)} = w_j^{(i)} U_j^{(i)}$ and $F^{(i)} = \sum_{1 \leqslant j \leqslant n_i} w_j^{(i)} U_j^{(i)}$. Under this assumption, the above minimality model is equivalent to the classical scheduling problem $1||\sum w_j U_j$, which has been well studied. Especially, when the weights of all the jobs are unit, Moore's algorithm [8] solves the problem in $O(n \log n)$ time. Hence, we study the feasibility model

$$1|| \sum_{1 \leqslant j \leqslant n_i} w_j^{(i)} U_j^{(i)} \leqslant Q_i, \quad 1 \leqslant i \leqslant m$$

in the following.

When $m = 2$ and $w_j^{(i)} = 1$ for each job $J_j^{(i)}$, by [1], the feasibility model $1||\sum_{1 \leqslant j \leqslant n_i} U_j^{(i)} \leqslant Q_i, 1 \leqslant i \leqslant m$, can be solved in polynomial time.

This paper seeks to extend the above result to a more general context. The idea for the algorithms in this paper partially comes from [1]. This paper is organized as follows. In Section 2 we present a simple approach that eliminates the agents i with $Q_i = 0$. In Section 3 we give an exact algorithm to solve the problem $1||\sum_{1 \leqslant j \leqslant n_i} w_j^{(i)} U_j^{(i)} \leqslant Q_i, 1 \leqslant i \leqslant m$, in general. It is shown that, when m is fixed, the algorithm runs in pseudo-polynomial time, and when m is fixed and $w_j^{(i)} = 1$ for each job $J_j^{(i)}$, the algorithm runs in polynomial time. In Section 4 we present a fully polynomial-time approximation scheme for the considered problem when m is fixed. In Section 5 we show that the problem $1||\sum_{1 \leqslant j \leqslant n_i} w_j^{(i)} U_j^{(i)} \leqslant Q_i, 1 \leqslant i \leqslant m$, is strongly NP-complete in general.

2. Eliminating the agents i with $Q_i = 0$

To save the computational effort, we first present an approach to eliminate the agents i with $Q_i = 0$. The following lemma can be observed.

Lemma 2.1. $1||\sum_{1 \leqslant j \leqslant n_i} w_j^{(i)} U_j^{(i)} \leqslant Q_i, 1 \leqslant i \leqslant m$, *is equivalent to*

$$1|pmtn| \sum_{1 \leqslant j \leqslant n_i} w_j^{(i)} U_j^{(i)} \leqslant Q_i, \quad 1 \leqslant i \leqslant m.$$

Suppose $Q_1 \geqslant Q_2 \geqslant \cdots \geqslant Q_m \geqslant 0$. Let $m' \leqslant m$ be the maximum such that $Q_{m'} > 0$. Without loss of generality, we suppose $m' < m$.

Write

$$\mathcal{J}^* = \mathcal{J}^{(m'+1)} \cup \mathcal{J}^{(m'+2)} \cup \cdots \cup \mathcal{J}^{(m)},$$

and suppose $|\mathcal{J}^*| = n^*$ and $\mathcal{J}^* = \{J_1, J_2, \ldots, J_{n^*}\}$. The due date of a job $J_j \in \mathcal{J}^*$ is denoted by D_j. Since all the jobs in $\mathcal{J}^*$ must be early in a feasible schedule, we call D_j the deadline of J_j.

We re-label the jobs in $\mathcal{J}^*$ in the earliest due date (EDD) order, i.e., $D_1 \leqslant D_2 \leqslant \cdots \leqslant D_{n^*}$. Then we define a sequence of numbers $(t_1, t_2, \ldots, t_{n^*})$ by the following dynamic programming recursion:

$$t_{n^*} = D_{n^*},$$
$$t_j = \min\{D_j, t_{j+1} - p_{j+1}\}, \quad j = n^* - 1, n^* - 2, \ldots, 1.$$

Clearly, the sequence $(t_1, t_2, \ldots, t_{n^*})$ can be obtained in $O(n^*)$ time.

Lemma 2.2. *If* $1|pmtn|\sum_{1 \leqslant j \leqslant n_i} w_j^{(i)} U_j^{(i)} \leqslant Q_i$, $1 \leqslant i \leqslant m$, *is feasible, then there is an optimal schedule* σ *such that the jobs in* $\mathcal{J}^*$ *are processed non-preemptively in non-decreasing order of their deadlines* (*EDD*), *and for each job* $J_j \in \mathcal{J}^*$, $1 \leqslant j \leqslant n^*$, *the time interval occupied by the job* J_j *under* σ *is exactly* $[t_j - p_j, t_j)$.

Proof. By Lemma 2.1, there must be a feasible schedule for $1|pmtn|\sum_{1 \leqslant j \leqslant n_i} w_j^{(i)} U_j^{(i)} \leqslant Q_i$, $1 \leqslant i \leqslant m$, such that (all the jobs, and so) the jobs in $\mathcal{J}^*$ are processed non-preemptively. Let $x \in \{0, 1, \ldots, n^*\}$ be the minimum such that there is a feasible schedule π for the problem, such that the jobs in $\mathcal{J}^*$ are processed non-preemptively and, for every job J_j with $x + 1 \leqslant j \leqslant n^*$, the time interval occupied by job J_j under σ is exactly $[t_j - p_j, t_j)$. We only need to show that $x = 0$.

Suppose to the contrary that $x > 0$. If there is some job J_y with $1 \leqslant y \leqslant x$ such that $C_y(\pi) > t_x$, then $D_x \geqslant D_y \geqslant C_y(\pi) > t_x$. By the definition of t_j, $1 \leqslant j \leqslant n^*$, we have $t_x = t_{x+1} - p_{x+1}$, where we assume $t_{n^*+1} = \sum_{1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n_i} p_j^{(i)}$ and $p_{n^*+1} = 0$. So, at least one job J_j with $j > x$ is processed before J_y. Suppose that the last of such jobs is J_z. Then, $D_z \geqslant D_y \geqslant C_y(\pi)$ and $t_{z+1} - p_{z+1} \geqslant C_y(\pi)$. Thus, we must have

$$t_z = \min\{D_z, t_{z+1} - p_{z+1}\} \geqslant C_y(\pi) > C_z(\pi) = t_z,$$

a contradiction. Hence, for every job J_y with $1 \leqslant y \leqslant x$, we must have $C_y(\pi) \leqslant t_x$. By shifting the processing of job J_x later to the interval $[t_x - p_x, t_x)$, we obtain another feasible schedule, which contradicts the choice of π. The result follows. $\square$

It should be pointed out that, by the above lemma, if $t_1 - p_1 < 0$, then the multi-agent scheduling problem has no feasible schedules.

By Lemma 2.2, we can assume that each job $J_j \in \mathcal{J}^*$, $1 \leqslant j \leqslant n^*$, has been processed in the time interval $[t_j - p_j, t_j)$ in advance. Follow the terminology in [9], the jobs in $\mathcal{J}^*$ are called fixed jobs. Then the remaining matter is to schedule the other jobs (called free jobs in the following) in $\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)} \cup \cdots \cup \mathcal{J}^{(m')}$ preemptively in the time space not occupied by the fixed jobs such that $\sum_{1 \leqslant j \leqslant n_i} w_j^{(i)} U_j^{(i)} \leqslant Q_i$, $1 \leqslant i \leqslant m'$. The corresponding problem is denoted by

$$1|FB, pmtn|\sum_{1 \leqslant j \leqslant n_i} w_j^{(i)} U_j^{(i)} \leqslant Q_i, \quad 1 \leqslant i \leqslant m'.$$

For each due date $d_j^{(i)}$ of a free job $J_j^{(i)}$, let $l_j^{(i)}$ be the sum of the length of the time slots occupied by the fixed jobs before the time instant $d_j^{(i)}$. Applying the same technique as in [11], we can without loss of generality delete the fixed jobs from consideration and modify the due dates in the following way:

$$d_j^{(i)} := d_j^{(i)} - l_j^{(i)}, \quad 1 \leqslant i \leqslant m, \ 1 \leqslant j \leqslant n_i.$$

After deleting the fixed jobs, preemption need not be considered. Hence, we have:

Lemma 2.3. *$1|FB, pmtn|\sum_{1\leqslant j\leqslant n_i} w_j^{(i)}U_j^{(i)} \leqslant Q_i$, $1\leqslant i\leqslant m'$, is reducible in linear time to the problem $1||\sum_{1\leqslant j\leqslant n_i} w_j^{(i)}U_j^{(i)} \leqslant Q_i$, $1\leqslant i\leqslant m'$, with modified due dates.*

The above discussion means that we can reduce in $O(n_1+n_2+\cdots+n_{m'}+n^*\log n^*)$ time the problem $1||\sum_{1\leqslant j\leqslant n_i} w_j^{(i)}U_j^{(i)} \leqslant Q_i$, $1\leqslant i\leqslant m$, to the problem $1||\sum_{1\leqslant j\leqslant n_i} w_j^{(i)}U_j^{(i)} \leqslant Q_i$, $1\leqslant i\leqslant m'$, in which $Q_i > 0$ for each agent.

3. An exact algorithm

In this section we will give an exact algorithm for the problem $1||\sum_{1\leqslant j\leqslant n_i} w_j^{(i)}U_j^{(i)} \leqslant Q_i$, $1\leqslant i\leqslant m$, when the weights of all the jobs and the values Q_i of all the agents are positive integers. By Lawler and Moore [7], this problem is binary NP-complete even when $m = 1$.

The following is an easy observation. The proof is the same as Lemma 7.1 in [1].

Lemma 3.1. *If the problem $1||\sum_{1\leqslant j\leqslant n_i} w_j^{(i)}U_j^{(i)} \leqslant Q_i$, $1\leqslant i\leqslant m$, is feasible, then there is a feasible schedule under which all the early jobs are scheduled consecutively in the EDD order at the beginning of the schedule.*

Now, suppose that the jobs are re-labelled in the EDD order, i.e., $\bigcup_{1\leqslant i\leqslant m}\mathcal{J}^{(i)} = \{J_1, J_2, \ldots, J_n\}$ such that $d_1\leqslant d_2\leqslant\cdots\leqslant d_n$. This needs $O(n\log n)$ time. Write $\mathcal{J}_k = \{J_1, J_2, \ldots, J_k\}$.

We consider the multi-agent scheduling problem $M(k; X_1, \ldots, X_m)$:

$$1||\sum_{j:J_j^{(i)}\in\mathcal{J}_k} w_j^{(i)}U_j^{(i)} \leqslant X_i, \quad 1\leqslant i\leqslant m,$$

subject to the jobs in $\mathcal{J}_k = \{J_1, J_2, \ldots, J_k\}$. Let $C(k; X_1, X_2, \ldots, X_m)$ be the minimum completion time of the last early job in a feasible schedule for the problem. If no feasible schedule exists, we set $C(k; X_1, X_2, \ldots, X_m) = +\infty$.

Let σ be a feasible schedule for the problem $M(k; X_1, \ldots, X_m)$ such that the completion time of the last early job is $C(k; X_1, X_2, \ldots, X_m)$. If J_k is an early job, it must be the last early job, and so we have $C(k; X_1, X_2, \ldots, X_m)\leqslant d_k$. In this case, we have

$$C(k; X_1, X_2, \ldots, X_m) = C(k-1; X_1, X_2, \ldots, X_m) + p_k.$$

If J_k is a tardy job and $J_k \in \mathcal{J}^{(i)}$, then we must have

$$C(k; X_1, X_2, \ldots, X_m) = C(k-1; X_1, \ldots, X_{i-1}, X_i - w_k, X_{i+1}, \ldots, X_m).$$

The above discussion implies the following dynamic programming recursion.

- Initial condition:

$$\begin{array}{ll} C(0 : X_1, \ldots, X_m) = 0 & \text{if all } X_i \geqslant 0, \\ C(k; X_1, X_2, \ldots, X_m) = +\infty & \text{if } k < 0 \text{ or some } X_i < 0 \text{ or some } X_i > Q_i. \end{array}$$

- Recursion function:

  If $J_k \in \mathcal{J}^{(i)}$ for some i with $1\leqslant i\leqslant m$, then

$$C(k; X_1, X_2, \ldots, X_m) = \min\begin{cases} C(k-1; X_1, X_2, \ldots, X_m) + p_k + f(k; X_1, X_2, \ldots, X_m), \\ C(k-1; X_1, \ldots, X_{i-1}, X_i - w_k, X_{i+1}, \ldots, X_m), \end{cases}$$

  where $f(k; X_1, X_2, \ldots, X_m) = 0$ if $C(k-1; X_1, X_2, \ldots, X_m) + p_k \leqslant d_k$, and $+\infty$ otherwise.

The above recursion function has $O(nQ_1Q_2, \ldots, Q_m)$ states. Each iteration needs constant computational time. The problem $1||\sum_{1\leqslant j\leqslant n_i} w_j^{(i)}U_j^{(i)} \leqslant Q_i$, $1\leqslant i\leqslant m$, is feasible if and only if $C(n; Q_1, Q_2, \ldots, Q_m) < +\infty$. We conclude the following result.

Theorem 3.2. *The problem $1||\sum_{1\leqslant j\leqslant n_i} w_j^{(i)}U_j^{(i)} \leqslant Q_i$, $1\leqslant i\leqslant m$, can be solved in $O((n_1+n_2+\cdots+n_m)Q_1Q_2, \ldots, Q_m)$ time. When m is fixed, it is pseudo-polynomial.*

When all the weights are unit, it is reasonable to assume $Q_i \leqslant n_i$. Hence, we further have:

Corollary 3.3. *The problem* $1||\sum_{1\leqslant j\leqslant n_i} U_j^{(i)} \leqslant Q_i$, $1\leqslant i\leqslant m$, *can be solved in* $\mathrm{O}((n_1+n_2+\cdots+n_m)n_1n_2,\ldots,n_m)$ *time. When m is fixed, it is polynomial.*

4. A fully polynomial-time approximation scheme

In this section we present an approximation algorithm for the problem

$$\mathcal{P} := 1||\sum_{1\leqslant j\leqslant n_i} w_j^{(i)} U_j^{(i)} \leqslant Q_i, \quad 1\leqslant i\leqslant m.$$

Suppose that the weights of all the jobs and the values Q_i of all the agents are positive numbers (but not necessarily integers). For any given constant $\varepsilon > 0$, the algorithm either finds a feasible schedule for the problem

$$\mathcal{P}' := 1||\sum_{1\leqslant j\leqslant n_i} w_j^{(i)} U_j^{(i)} \leqslant (1+\varepsilon) Q_i, \quad 1\leqslant i\leqslant m,$$

or determines that problem $\mathcal{P}$ has no feasible schedule. Note that problem $\mathcal{P}'$ is a relaxation of problem $\mathcal{P}$ in which each Q_i is enlarged by a factor of $(1+\varepsilon)$.

Let $\varepsilon > 0$ be a given constant. The weights of all the jobs $J_j^{(i)}$, $1\leqslant j\leqslant n_i$, $1\leqslant i\leqslant m$, are rounded in the following way:

$$v_j^{(i)} = \begin{cases} \left\lceil \dfrac{2n_i}{\varepsilon} \right\rceil + n_i + 1 & \text{if } w_j^{(i)} > Q_i, \\ \left\lceil \dfrac{2n_i w_j^{(i)}}{\varepsilon Q_i} \right\rceil & \text{otherwise.} \end{cases}$$

The threshold values Q_i, $1\leqslant i\leqslant m$, are rounded in the following way:

$$Q_i^* = \left\lceil \frac{2n_i}{\varepsilon} \right\rceil + n_i.$$

We define problem $\mathcal{P}^*$ as

$$1||\sum_{1\leqslant j\leqslant n_i} v_j^{(i)} U_j^{(i)} \leqslant Q_i^*, \quad 1\leqslant i\leqslant m.$$

Theorem 4.1. *If problem $\mathcal{P}^*$ has a feasible schedule π, then π is a feasible schedule for problem $\mathcal{P}'$; otherwise, problem $\mathcal{P}$ has no feasible schedule.*

Proof. If problem $\mathcal{P}^*$ has a feasible schedule π, then

$$\sum_{1\leqslant j\leqslant n_i} v_j^{(i)} U_j^{(i)}(\pi) \leqslant Q_i^*, \quad 1\leqslant i\leqslant m.$$

By the definition of $v_j^{(i)}$ and Q_i^*, we see that the following three statements hold:

(1) if $w_j^{(i)} > Q_i$, then $J_j^{(i)}$ is an early job in π;

(2) if $w_j^{(i)} \leqslant Q_i$, then

$$w_j^{(i)} \leqslant v_j^{(i)} (\varepsilon Q_i / 2n_i);$$

(3) $Q_i^*(\varepsilon Q_i/2n_i) \leqslant (1+\varepsilon) Q_i$.

Hence, for each agent i,

$$\begin{aligned}
&\sum_{1\leqslant j\leqslant n_i} w_j^{(i)}U_j^{(i)}(\pi)\\
&\quad\leqslant(\varepsilon Q_i/2n_i)\sum_{1\leqslant j\leqslant n_i} v_j^{(i)}U_j^{(i)}(\pi)\\
&\quad\leqslant(\varepsilon Q_i/2n_i)Q_i^*\\
&\quad\leqslant(1+\varepsilon)Q_i.
\end{aligned}$$

It follows that π is a feasible solution for problem $\mathcal{P}'$.

On the other hand, if problem $\mathcal{P}$ has a feasible schedule h, then, for each agent i,

$$\sum_{1\leqslant j\leqslant n_i} w_j^{(i)}U_j^{(i)}(h)\leqslant Q_i,$$

and so

$$\sum_{1\leqslant j\leqslant n_i} (2n_i w_j^{(i)}/\varepsilon Q_i)U_j^{(i)}(h)\leqslant 2n_i/\varepsilon.$$

Since $v_j^{(i)}\leqslant 2n_i w_j^{(i)}/\varepsilon Q_i+1$, we deduce that

$$\sum_{1\leqslant j\leqslant n_i} v_j^{(i)}U_j^{(i)}(h)\leqslant 2n_i/\varepsilon+n_i\leqslant Q_i^*$$

for each agent i. It follows that h is a feasible schedule for problem $\mathcal{P}^*$. The result follows. □

Since $Q_i^*=\mathrm{O}(2n_i/\varepsilon)$ for each agent i, by the discussion of the last section, problem $\mathcal{P}^*$ can be solved in $\mathrm{O}((n_1+n_2+\cdots+n_m)n_1n_2\ldots n_m(2/\varepsilon)^m)$ time. Consequently, we have the following result.

Theorem 4.2. *For any given constant $\varepsilon>0$, there is a polynomial-time algorithm running in $\mathrm{O}((n_1+n_2+\cdots+n_m)n_1n_2\ldots n_m(2/\varepsilon)^m)$ time that either finds a feasible schedule for problem $1||\sum_{1\leqslant j\leqslant n_i} w_j^{(i)}U_j^{(i)}\leqslant(1+\varepsilon)Q_i$, $1\leqslant i\leqslant m$, or determines that the problem $1||\sum_{1\leqslant j\leqslant n_i} w_j^{(i)}U_j^{(i)}\leqslant Q_i$, $1\leqslant i\leqslant m$, has no feasible schedule.*

When m is a fixed constant, Theorem 4.2 implies that, for the problem $1||\sum_{1\leqslant j\leqslant n_i} w_j^{(i)}U_j^{(i)}\leqslant Q_i$, $1\leqslant i\leqslant m$, there is a fully polynomial-time approximation scheme running in $\mathrm{O}((n_1+n_2+\cdots+n_m)n_1n_2\ldots n_m(1/\varepsilon)^m)$ time for any $\varepsilon>0$.

5. Strong NP-completeness

We show in this section that the problem $1||\sum_{1\leqslant j\leqslant n_i} w_j^{(i)}U_j^{(i)}\leqslant Q_i$, $1\leqslant i\leqslant m$, is strongly NP-complete when m is arbitrary.

Theorem 5.1. $1||\sum_{1\leqslant j\leqslant n_i} w_j^{(i)}U_j^{(i)}\leqslant Q_i$, $1\leqslant i\leqslant m$, *is strongly NP-complete.*

Proof. By Garey and Johnson [4], the 3-partition problem is strongly NP-complete. In an instance I of the 3-partition problem, we are given a set of $3t$ positive integers $a_1,a_2,\ldots,a_{3t}$, each of size between $B/4$ and $B/2$, such that $\sum_{i=1}^{3t}a_i=tB$. The decision asks whether there is a partition of the a_i's into t groups of 3, each summing exactly to B?

Given an instance I of the 3-partition problem, we first re-label the $3t$ numbers in I such that $a_1\leqslant a_2\leqslant\cdots\leqslant a_{3t}$. We construct an instance I^* of the scheduling problem $1||\sum_{1\leqslant j\leqslant n_i} w_j^{(i)}U_j^{(i)}\leqslant Q_i$, $1\leqslant i\leqslant m$, as follows:

- t agents and $3t^2$ jobs with each agent having exactly $3t$ jobs, i.e.,

$$\mathcal{J}^{(i)}=\left\{J_1^{(i)},J_2^{(i)},\ldots,J_{3t}^{(i)}\right\},\quad 1\leqslant i\leqslant t.$$

- Processing times of the jobs are defined by

$$p_j^{(i)} = X + a_j, \quad 1 \leqslant i \leqslant t, \ 1 \leqslant j \leqslant 3t,$$

where $X = 3t^2 B$.
- Due dates of the jobs are defined by

$$d_j^{(i)} = (t-1)Xj + (t-1)A_j, \quad 1 \leqslant i \leqslant t, \ 1 \leqslant j \leqslant 3t,$$

where $A_j = a_1 + a_2 + \cdots + a_j$.
- The weights of jobs are defined by

$$w_j^{(i)} = X + a_j, \quad 1 \leqslant i \leqslant t, \ 1 \leqslant j \leqslant 3t.$$

- Threshold values are defined by $Q_i = 3X + B$, $1 \leqslant i \leqslant t$.
- The decision asks whether there is a schedule π such that $\sum_{1 \leqslant j \leqslant 3t} w_j^{(i)} U_j^{(i)}(\pi) \leqslant 3X + B$ for each i with $1 \leqslant i \leqslant t$.

If I has a solution, then there is a t-partition $I_1, I_2, \ldots, I_t$ of $\{1, 2, \ldots, 3t\}$ (i.e., $I_1 \cup I_2 \cup \cdots \cup I_t = \{1, 2, \ldots, 3t\}$ and $I_i \cap I_j = \emptyset$ for $i \neq j$) such that $|I_i| = 3$ and $\sum_{j \in I_i} a_j = B$ for each i with $1 \leqslant i \leqslant t$. Define

$$\mathcal{U} = \{J_j^{(i)} : j \in I_i, 1 \leqslant i \leqslant t\}.$$

Let $\mathcal{J}$ be the set of all the jobs. The job subset $\mathcal{J} \setminus \mathcal{U}$ has the property that, for each j with $1 \leqslant j \leqslant 3t$, the total length of the jobs with a due date at most $(t-1)Xj + (t-1)A_j$ is exactly $(t-1)Xj + (t-1)A_j$. Hence, under the EDD sequence, every job in $\mathcal{J} \setminus \mathcal{U}$ is early. The total weight of the ith agent's jobs in $\mathcal{U}$ is

$$s \sum_{j \in I_i} w_j^{(i)} = \sum_{j \in I_i} (X + a_j) = 3X + B = Q_i.$$

Hence, an EDD sequence of the jobs in $\mathcal{J} \setminus \mathcal{U}$, followed by an arbitrary sequence of the jobs in $\mathcal{U}$, is a feasible schedule for I^*.

Conversely, suppose that there is a feasible schedule π for I^*. We can assume without loss of generality that the early jobs are scheduled in the EDD sequence before the processing of any tardy job. The set of all the tardy jobs under π is denoted by $\mathcal{U}$. For each job J, $p(J)$ is used to denote the processing time of J and $w(J)$ the weight of J.

Claim 1. $|\mathcal{U}| = 3t$.

Note that, for each job J, we have $X < p(J) = w(J) < X + B$. If $|\mathcal{U}| \geqslant 3t + 1$, then

$$\sum_{j \in \mathcal{U}} w(J) > (3t+1)X > Q_1 + Q_2 + \cdots + Q_t,$$

exceeding the threshold values, which is a contradiction. If $|\mathcal{U}| \leqslant 3t - 1$, then, by noting that the total processing time of the jobs is $3t^2 X + t^2 B$, we have

$$\sum_{j \in \mathcal{U}} p(J) < (3t-1)X + (3t-1)B < 3tX + tB = 3tX + A_{3t},$$

and so the maximum completion time of the early job is greater than $3t(t-1)X + (t-1)A_{3t}$, the last due date of the jobs. This contradicts the definition of $\mathcal{U}$. We conclude that $|\mathcal{U}| = 3t$. The proof of Claim 1 is completed.

Suppose that $(v_0, v_1, \ldots, v_k)$ is the unique index sequence such that $0 = v_0 < v_1 < v_2 < \cdots < v_k = 3t$,

$$a_{v_i+1} = a_{v_i+2} = \cdots = a_{v_{i+1}}, \quad i = 0, 1, \ldots, k-1$$

and

$$a_1 = a_{v_1} < a_{v_2} < \cdots < a_{v_k} = a_{3t}.$$

For each u with $1 \leqslant u \leqslant 3t$, set $\mathcal{J}_u = \{J_j^{(i)} : 1 \leqslant i \leqslant t, 1 \leqslant j \leqslant u\}$, $\mathcal{N}_u = \mathcal{U} \cap \mathcal{J}_u$ and $N_u = |\mathcal{N}_u|$.

Claim 2. $N_{v_s} = v_s$ for $s = 1, 2, \ldots, k$.

If $N_{v_s} < v_s$ for some s with $1 \leqslant s \leqslant k-1$, then the total length of the jobs in $\mathcal{N}_{v_s}$ is less than $(v_s - 1)X + (v_s - 1)B < v_s X + A_{v_s}$. Hence, the maximum completion time of the early jobs in $\mathcal{J}_{v_s}$ is greater than $(t-1)v_s X + (t-1)A_{v_s}$, the maximum due date of the jobs in $\mathcal{J}_{v_s}$, contradicting the definition of $\mathcal{N}_{v_s}$. Consequently, $N_{v_s} \geqslant v_s$ for $s = 1, 2, \ldots, k$. We notice that $N_{v_k} = |\mathcal{U}| = 3t = v_k$ always holds.

If $N_{v_s} > v_s$ for some s with $1 \leqslant s \leqslant k-1$, then the total length of all the tardy jobs minus $3tX$ can be calculated by

$$\begin{aligned}
&\textstyle\sum_{J \in \mathcal{U}} p(J) - 3tX \\
&\quad = \sum_{1 \leqslant s \leqslant k} (N_{v_s} - N_{v_{s-1}}) a_{v_s} \quad (\text{here, } N_{v_0} = 0) \\
&\quad = \sum_{1 \leqslant s \leqslant k} (v_s - v_{s-1}) a_{v_s} + \sum_{1 \leqslant s \leqslant k} ((N_{v_s} - v_s) - (N_{v_{s-1}} - v_{s-1})) a_{v_s} \\
&\quad = \sum_{1 \leqslant j \leqslant 3t} a_i + \textstyle\sum_{1 \leqslant s \leqslant k-1} (N_{v_s} - v_s)(a_{v_s} - a_{v_{s+1}}) \\
&\quad < \sum_{1 \leqslant j \leqslant 3t} a_i = A_{3t}.
\end{aligned}$$

That is, $\sum_{J \in \mathcal{U}} p(J) < 3tX + A_{3t}$. Consequently, the maximum completion time of the early jobs is greater than $3t(t-1)X + (t-1)A_{3t}$, the last due date of jobs. This contradicts the definition of $\mathcal{U}$.

We conclude that $N_{v_s} = v_s$ for $s = 1, 2, \ldots, k$. The proof of Claim 2 is completed.

By Claim 2, for each s with $1 \leqslant s \leqslant k$, there are exactly $t_s = N_{v_s} - N_{v_{s-1}} = v_s - v_{s-1}$ tardy jobs in $\mathcal{J}_{v_s} \setminus \mathcal{J}_{v_{s-1}}$ with a common processing time $X + a_s$. We construct a new set $\mathcal{U}^*$ in the following way:

(1) For s from 1 to k we do the following: Sequencing the t_s tardy jobs in $\mathcal{J}_{v_s} \setminus \mathcal{J}_{v_{s-1}}$ in an arbitrary order, say, $J_{s,1}, J_{s,2}, \ldots, J_{s,t_s}$. For each j with $1 \leqslant j \leqslant t_s$, let $x(s,j)$ be the agent index such that $J_{s,j} \in \mathcal{J}^{(x(s,j))}$. We notice that $J_{s,j}$ and $J^{(x(s,j))}_{v_{s-1}+j}$ belong to the same agent and have the same processing time and the same weight. Define $\mathcal{U}^*_s = \{J^{(x(s,j))}_{v_{s-1}+j} : 1 \leqslant j \leqslant t_s\}$.

(2) Set $\mathcal{U}^* = \bigcup_{1 \leqslant s \leqslant k} \mathcal{U}^*_s$.

By the construction of $\mathcal{U}^*$, we have

$$Y_i := \sum_{J \in \mathcal{U} \cap \mathcal{J}^{(i)}} w(J) = \sum^{*}_{J \in \mathcal{U} \cap \mathcal{J}^{(i)}} w(J) \leqslant Q_i = 3X + B$$

for each i with $1 \leqslant i \leqslant t$. For each u with $1 \leqslant u \leqslant 3t$, the job set $\mathcal{J}_u \setminus \mathcal{J}_{u-1}$ contains exactly one job in $\mathcal{U}^*$, where $\mathcal{J}_0 = \emptyset$. It follows that

$$\sum_{1 \leqslant i \leqslant t} Y_i = 3tX + A_{3t} = 3tX + tB.$$

Together with the fact that $Y_i \leqslant 3X + B$ for $1 \leqslant i \leqslant t$, we deduce that $Y_1 = Y_2 = \cdots = Y_t = 3x + B$. By setting

$$I_i = \{j : J_j^{(i)} \in \mathcal{U}^*,\ 1 \leqslant i \leqslant t\},$$

we have

$$|I_i| = 3 \quad \text{and} \quad \sum_{j \in I_i} a_j = B \text{ for each } i \text{ with } 1 \leqslant i \leqslant t.$$

We conclude that the instance I of 3-partition has a solution, too. The result follows. □

When m is arbitrary, it is still open whether the problem $1||\sum_{1 \leqslant j \leqslant n_i} U_j^{(i)} \leqslant Q_i$, $1 \leqslant i \leqslant m$, is polynomially solvable.

6. Conclusions

In this paper we studied the feasibility model of multi-agent scheduling on a single machine, where each agent's objective function is to minimize the total weighted number of tardy jobs. We showed that the problem is strongly

NP-complete in general. When the number of agents is fixed, we first showed that the problem can be solved in pseudo-polynomial time for integral weights, and can be solved in polynomial time in unit weights; then we presented a fully polynomial-time approximation scheme for the problem. For the future research, the complexity of the problem $1||\sum_{1\leqslant j\leqslant n_i} U_j^{(i)} \leqslant Q_i$, $1\leqslant i\leqslant m$, is still open when m is arbitrary. The approximation algorithm for the problem $1||\sum_{1\leqslant j\leqslant n_i} w_j^{(i)} U_j^{(i)} \leqslant Q_i$, $1\leqslant i\leqslant m$, also needs to be resolved when m is arbitrary.

Acknowledgements

This research was supported in part by The Hong Kong Polytechnic University under grant number S818. The third author was also supported in part by NSFC(10371112), NSFHN (0411011200) and SRF for ROCS, SEM.

References

[1] A. Agnetis, P.B. Mirchandani, D. Pacciarelli, A. Pacifici, Scheduling problems with two competing agents, Oper. Res. 52 (2004) 229–242.
[2] K.R. Baker, J.C. Smith, A multiple-criterion model for machine scheduling, J. Scheduling 6 (2003) 7–16.
[3] I. Curiel, G. Pederzoli, S. Tijs, Sequencing games, European J. Oper. Res. 40 (1989) 344–351.
[4] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.
[5] H. Hamers, P. Borm, S. Tijs, On games corresponding to sequencing situations with ready times, Math. Programming 70 (1995) 1–13.
[6] K. Kim, B.C. Paulson, C.J. Petrie, V.R. Lesser, Compensatory negotiation for agent-based project schedule coordination, CIFE Working Paper No. 55, Stanford University, Stanford, CA, 1999.
[7] E.L. Lawler, J.M. Moore, A functional equation and its application to resource allocation and sequencing problems, Management Sci. 16 (1969) 77–84.
[8] J.M. Moore, An n job, one machine sequencing algorithm for minimizing the number of late jobs, Management Sci. 15 (1968) 102–109.
[9] M. Scharbrodt, A. Steger, H. Weisser, Approximability of scheduling with fixed jobs, J. Scheduling 2 (1999) 267–284.
[10] D. Schultz, S.-H. Oh, C.F. Grecas, M. Albani, J. Sanchez, C. Arbib, V. Arvia, M. Servilio, F. Del Sorbo, A. Giralda, G. Lombardi, A QoS concept for packet oriented S-UMTS services, In: Proc. of the 1st Mobile Summit 2002, Thessaloniki, Greece.
[11] J.J. Yuan, Y.X. Lin, Single machine preemptive scheduling with fixed jobs to minimize tardiness related criteria, European J. Oper. Res. 164 (2005) 851–858.