

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

T.C.E. Cheng<sup>1\*</sup>, C.T. Ng<sup>1</sup> and J.J. Yuan<sup>2</sup>

<sup>1</sup>Department of Logistics, The Hong Kong Polytechnic University,  
Hung Hom, Kowloon, Hong Kong, People's Republic of China

<sup>2</sup>Department of Mathematics, Zhengzhou University,  
Zhengzhou, Henan 450052, People's Republic of China

**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. (2004) and Baker and Smith (2003). 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, Curiel et al. (1989) and Hamers et al.

---

\*Corresponding author

(1995)). In project scheduling, the problem is concerned with negotiation to resolve conflicts whenever the agents find their own schedules unacceptable (Kim et al. (1999)). 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 (Schultz et al. (2002)).

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  $\mathcal{J}^{(1)}, \mathcal{J}^{(2)}, \dots, \mathcal{J}^{(m)}$ , where, for each  $i$  with  $1 \leq i \leq m$ ,

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

The jobs in  $\mathcal{J}^{(i)}$  are called the  $i$ -th 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  $\sigma$ , 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 nondecreasing function of the completion time of job  $J_j^{(i)}$  (such an objective function is called *regular* in the scheduling literature).

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

$$\begin{aligned} \text{max-form} \quad & F^{(i)}(\sigma) = \max_{1 \leq j \leq n_i} f_j^{(i)}(C_j^{(i)}(\sigma)), \\ \text{sum-form} \quad & F^{(i)}(\sigma) = \sum_{1 \leq j \leq n_i} f_j^{(i)}(C_j^{(i)}(\sigma)). \end{aligned}$$

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

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

In this paper we always assume that  $f_j^{(i)} = w_j^{(i)}U_j^{(i)}$  and  $F^{(i)} = \sum_{1 \leq j \leq n_i} w_j^{(i)}U_j^{(i)}$ . Under this assumption, the above minimality model is equivalent to the classical scheduling problem  $1 \parallel \sum w_j U_j$ , which has been well studied. Especially, when the weights of all The jobs are unit, Moore's algorithm (Moore, 1968) solves the problem in  $O(n \log n)$  time. Hence, we study the feasibility model

$$1 \parallel \sum_{1 \leq j \leq n_i} w_j^{(i)} U_j^{(i)} \leq Q_i, 1 \leq i \leq m$$

in the following.

When  $m = 2$  and  $w_j^{(i)} = 1$  for each job  $J_j^{(i)}$ , by Agnetis et al. (2004), the feasibility model  $1 \parallel \sum_{1 \leq j \leq n_i} U_j^{(i)} \leq Q_i, 1 \leq i \leq m$  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 Agnetis et al. (2004). 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  $1 \parallel \sum_{1 \leq j \leq n_i} w_j^{(i)} U_j^{(i)} \leq Q_i, 1 \leq i \leq m$  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  $1 \parallel \sum_{1 \leq j \leq n_i} w_j^{(i)} U_j^{(i)} \leq Q_i, 1 \leq i \leq m$  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**  $1 \parallel \sum_{1 \leq j \leq n_i} w_j^{(i)} U_j^{(i)} \leq Q_i, 1 \leq i \leq m$  is equivalent to

$$1 \parallel \sum_{1 \leq j \leq n_i} w_j^{(i)} U_j^{(i)} \leq Q_i, 1 \leq i \leq m.$$

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

Write

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

and suppose  $|\mathcal{J}^*| = n^*$  and  $\mathcal{J}^* = \{J_1, J_2, \dots, J_{n^*}\}$ . 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.,  $D_1 \leq D_2 \leq \dots \leq D_{n^*}$ . Then we define a sequence of numbers  $(t_1, t_2, \dots, t_{n^*})$  by the following dynamic programming recursion:

$$\begin{aligned} t_{n^*} &= D_{n^*}, \\ t_j &= \min\{D_j, t_{j+1} - p_{j+1}\}, \quad j = n^* - 1, n^* - 2, \dots, 1. \end{aligned}$$

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

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

**Proof** By Lemma 2.1, there must be a feasible schedule for  $1|pmtn|\sum_{1 \leq j \leq n_i} w_j^{(i)} U_j^{(i)} \leq Q_i, 1 \leq i \leq m$  such that (all the jobs, and so) the jobs in  $\mathcal{J}^*$  are processed non-preemptively. Let  $x \in \{0, 1, \dots, n^*\}$  be the minimum such that there is a feasible schedule  $\pi$  for the problem, such that the jobs in  $\mathcal{J}^*$  are processed non-preemptively and, for every job  $J_j$  with  $x + 1 \leq j \leq n^*$ , the time interval occupied by job  $J_j$  under  $\sigma$  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  $1 \leq y \leq x$  such that  $C_y(\pi) > t_x$ , then  $D_x \geq D_y \geq C_y(\pi) > t_x$ . By the definition of  $t_j, 1 \leq j \leq n^*$ , we have  $t_x = t_{x+1} - p_{x+1}$ , where we assume  $t_{n^*+1} = \sum_{1 \leq i \leq m, 1 \leq j \leq n_i} p_j^{(i)}$  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,  $D_z \geq D_y \geq C_y(\pi)$  and  $t_{z+1} - p_{z+1} \geq C_y(\pi)$ . Thus, we must have

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

a contradiction. Hence, for every job  $J_y$  with  $1 \leq y \leq x$ , we must have  $C_y(\pi) \leq t_x$ . 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  $\pi$ . 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  $J_j \in \mathcal{J}^*, 1 \leq j \leq n^*$ , has been processed in the time interval  $[t_j - p_j, t_j]$  in advance. Follow the terminology in Scharbrodt et al. (1999), 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  $\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)} \cup \dots \cup \mathcal{J}^{(m')}$  preemptively in the time space not occupied by the fixed jobs such that  $\sum_{1 \leq j \leq n_i} w_j^{(i)} U_j^{(i)} \leq Q_i, 1 \leq i \leq m'$ . The corresponding problem is denoted by

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

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 Yuan and Lin (2005), we can without loss of generality delete the fixed jobs from consideration and modify the due dates in the following way:

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

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

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

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

### 3 An exact algorithm

In this section we will give an exact algorithm for the problem  $1|| \sum_{1 \leq j \leq n_i} w_j^{(i)} U_j^{(i)} \leq Q_i, 1 \leq i \leq m$  when the weights of all the jobs and the values  $Q_i$  of all the agents are positive integers. By Lawler and Moore (1969), 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 Agnetis et al. (2004).

**Lemma 3.1** If the problem  $1|| \sum_{1 \leq j \leq n_i} w_j^{(i)} U_j^{(i)} \leq Q_i, 1 \leq i \leq m$  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.,  $\cup_{1 \leq i \leq m} \mathcal{J}^{(i)} = \{J_1, J_2, \dots, J_n\}$  such that  $d_1 \leq d_2 \leq \dots \leq d_n$ . This needs  $O(n \log n)$  time. Write  $\mathcal{J}_k = \{J_1, J_2, \dots, J_k\}$ .

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

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

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

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

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

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

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

The above discussion implies the following dynamic programming recursion.

- Initial condition:

$$C(0 : X_1, \dots, X_m) = 0, \quad \text{if all } X_i \geq 0,$$

$$C(k; X_1, X_2, \dots, X_m) = +\infty, \quad \text{if } k < 0 \text{ or some } X_i < 0 \text{ or some } X_i > Q_i.$$

- Recursion function:

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

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

where  $f(k; X_1, X_2, \dots, X_m) = 0$  if  $C(k; X_1, X_2, \dots, X_m) \leq d_k$ , and  $+\infty$  otherwise.

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

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

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

**Corollary 3.3** The problem  $1 \parallel \sum_{1 \leq j \leq n_i} U_j^{(i)} \leq Q_i, 1 \leq i \leq m$  can be solved in  $O((n_1 + n_2 + \dots + n_m)n_1n_2\dots n_m)$  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

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

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

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

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  $J_j^{(i)}$ ,  $1 \leq j \leq n_i$ ,  $1 \leq i \leq m$ , are rounded in the following way:

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

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

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

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

$$1 \parallel \sum_{1 \leq j \leq n_i} v_j^{(i)} U_j^{(i)} \leq Q_i^*, 1 \leq i \leq m.$$

**Theorem 4.1** If problem  $\mathcal{P}^*$  has a feasible schedule  $\pi$ , then  $\pi$  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  $\pi$ , then

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

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  $\pi$ ;
- (2) If  $w_j^{(i)} \leq Q_i$ , then

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

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

Hence, for each agent  $i$ ,

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

It follows that  $\pi$  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  $i$ ,

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

and so

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

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

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

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

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

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

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

## 5 Strong NP-completeness

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

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

**Proof** By Garey and Johnson (1979), 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  $a_1, a_2, \dots, a_{3t}$ , 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  $a_1 \leq a_2 \leq \dots \leq a_{3t}$ . We construct an instance  $I^*$  of the scheduling problem  $1 \parallel \sum_{1 \leq j \leq n_i} w_j^{(i)} U_j^{(i)} \leq Q_i, 1 \leq i \leq m$  as follows.

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

$$\mathcal{J}^{(i)} = \{J_1^{(i)}, J_2^{(i)}, \dots, J_{3t}^{(i)}\}, \quad 1 \leq i \leq t.$$



- Processing times of the jobs are defined by

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

where  $X = 3t^2B$ .

- Due dates of the jobs are defined by

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

where  $A_j = a_1 + a_2 + \dots + a_j$ .

- The weights of jobs are defined by

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

- Threshold values are defined by  $Q_i = 3X + B$ ,  $1 \leq i \leq t$ .

- The decision asks whether there is a schedule  $\pi$  such that  $\sum_{1 \leq j \leq 3t} w_j^{(i)} U_j^{(i)}(\pi) \leq 3X + B$  for each  $i$  with  $1 \leq i \leq t$ .

If  $I$  has a solution, then there is a  $t$ -partition  $I_1, I_2, \dots, I_t$  of  $\{1, 2, \dots, 3t\}$  (i.e.,  $I_1 \cup I_2 \cup \dots \cup I_t = \{1, 2, \dots, 3t\}$  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  $1 \leq i \leq t$ . Define

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

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  $1 \leq j \leq 3t$ , 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  $i$ -th agent's jobs in  $\mathcal{U}$  is

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

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  $\pi$  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  $\pi$  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  $|\mathcal{U}| \geq 3t + 1$ , then

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

exceeding the threshold values, which is a contradiction. If  $|\mathcal{U}| \leq 3t - 1$ , then, by noting that the total processing time of the jobs is  $3t^2X + t^2B$ , 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  $(v_0, v_1, \dots, v_k)$  is the unique index sequence such that  $0 = v_0 < v_1 < v_2 < \dots < v_k = 3t$ ,

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

and

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

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

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

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

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

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

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  $s = 1, 2, \dots, k$ . The proof of Claim 2 is completed.

By Claim 2, for each  $s$  with  $1 \leq s \leq k$ , 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,  $J_{s,1}, J_{s,2}, \dots, J_{s,t_s}$ . For each  $j$  with  $1 \leq j \leq t_s$ , 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  $J_{v_{s-1}+j}^{(x(s,j))}$  belong to the same agent and have the same processing time and the same weight. Define  $\mathcal{U}_s^* = \{J_{v_{s-1}+j}^{(x(s,j))} : 1 \leq j \leq t_s\}$ .

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

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

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

for each  $i$  with  $1 \leq i \leq t$ . For each  $u$  with  $1 \leq u \leq 3t$ , 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

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

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

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

we have

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

We conclude that the instance  $I$  of 3-Partition has a solution, too. The result follows.  $\square$

When  $m$  is arbitrary, it is still open whether the problem  $1|| \sum_{1 \leq j \leq n_i} U_j^{(i)} \leq Q_i, 1 \leq i \leq m$  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  $1|| \sum_{1 \leq j \leq n_i} U_j^{(i)} \leq Q_i, 1 \leq i \leq m$  is still open when  $m$  is arbitrary. The approximation algorithm for the problem  $1|| \sum_{1 \leq j \leq n_i} w_j^{(i)} U_j^{(i)} \leq Q_i, 1 \leq i \leq m$  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

- Agnētis, A., P.B. Mirchandani, D. Pacciarelli, and A. Pacifici, “Scheduling problems with two competing agents”, *Operations Research*, **52**, 229-242 (2004).
- Baker, K.R., and J.C. Smith, “A multiple-criterion model for machine scheduling”, *Journal of Scheduling*, **6**, 7-16 (2003).
- Curiel, I., G. Pederzoli, and S. Tijs, “Sequencing games”, *European Journal of Operational Research*, **40**, 344-351 (1989).
- Garey, M.R., and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, 1979.
- Hamers, H., P. Borm, and S. Tijs, “On games corresponding to sequencing situations with ready times”, *Mathematical Programming*, **70**, 1-13 (1995).
- Kim, K., B.C. Paulson, C.J. Petrie, and V.R. Lesser, “Compensatory negotiation for agent-based project schedule coordination”, CIFE working paper #55, Stanford University, Stanford, CA, 1999
- Lawler, E.L., and J.M. Moore, “A functional equation and its application to resource allocation and sequencing problems”, *Management Science*, **16**, 77-84 (1969).
- Moore, J.M., “An  $n$  job, one machine sequencing algorithm for minimizing the number of late jobs”, *Management Science*, **15**, 102-109 (1968).
- Scharbrodt, M., A. Steger, and H. Weisser, “Approximability of scheduling with fixed jobs”, *Journal of Scheduling*, **2**, 267-284 (1999).
- Schultz, D., S.-H. Oh, C.F. Grecas, M. Albani, J. Sanchez, C. Arbib, V. Arvia, M. Servilio, F. Del Sorbo, A. Giralda, and G. Lombardi, “A QoS concept for packet oriented S-UMTS services”, *Proceedings of the 1st Mobile Summit 2002*, Thessaloniki, Greece.
- Yuan, J.J. and Y.X. Lin, “Single machine preemptive scheduling with fixed jobs to minimize tardiness related criteria”, *European Journal of Operational Research*, **164**, 851-858 (2005).