Journal of Scheduling **6:** 405–412, 2003.

CONCURRENT OPEN SHOP SCHEDULING TO MINIMIZE THE WEIGHTED NUMBER OF TARDY JOBS

C. T. NG[1,*], T. C. E. CHENG[1], AND J. J. YUAN[1,2]

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

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

ABSTRACT

We consider a relaxed version of the open shop scheduling problem—the "concurrent open shop" scheduling problem, in which any two operations of the same job on distinct machines are allowed to be processed concurrently. The completion time of a job is the maximum completion time of its operations. The objective is to schedule the jobs so as to minimize the weighted number of tardy jobs, with 0–1 operation processing times and a common due date d. We show that, even when the weights are identical, the problem has no $(1-\varepsilon)\ln m$-approximation algorithm for any $\epsilon > 0$ if NP is not a subset of DTIME($n^{\log\log n}$), and has no $c \cdot \ln m$-approximation algorithm for some constant $c > 0$ if $P \neq NP$, where m is the number of machines. This also implies that the problem is strongly NP-hard. We also give a $(1+d)$-approximation algorithm for the problem.

KEY WORDS: scheduling; concurrent open shop; due date; in-approximability

1. INTRODUCTION AND PROBLEM FORMULATION

The concurrent open shop scheduling problem can be stated as follows: Let n jobs

$$J_1, J_2, \ldots, J_n$$

and m machines

$$M_1, M_2, \ldots, M_m$$

be given. Each job J_j consists of m independent operations

$$O_{1,j}, O_{2,j}, \ldots, O_{m,j}$$

An operation $O_{i,j}$ is to be processed on machine M_i, $i = 1, 2, \ldots, m$. The processing time of $O_{i,j}$ is denoted by $p_{i,j}$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$. The completion time of a job is the maximum completion time of its operations. d_j is the due date of job J_j. The weight of job J_j is denoted by w_j (≥ 0). The objective is to schedule the jobs to minimize the weighted number of tardy jobs.

* Corresponding author: C. T. Ng, Department of Management, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, People's Republic of China.

Contract/grant sponsor: Research Grant Council of Hong Kong; Contract/grant number: PolyU 5198/02E

For a given schedule π, we will use the following notation:

$C_{i,j}(\pi)$, the completion time of operation $O_{i,j}$
$C_j(\pi) = \max_{1 \leq i \leq m} C_{i,j}(\pi)$, the completion time of job J_j
$U_j(\pi) = 1$, if $C_j(\pi) > d_j$, and 0, otherwise
$\sum_{1 \leq j \leq n} U_j(\pi)$, the number of tardy jobs
$\sum_{1 \leq j \leq n} w_j U_j(\pi)$, the weighted number of tardy jobs.

When the number m of machines is fixed, the scheduling problem considered is denoted by

$$Mm||\sum w_j U_j$$

and when the number m of machines is a part of the input to the problem, the scheduling problem considered is denoted by

$$M||\sum w_j U_j$$

One can easily see that the concurrent open shop scheduling problem is equivalent to the customer order scheduling problem (Cheng and Wang, 1999), in which each customer order (jobs) consists of several jobs (operations) that have to be processed by different facilities (machines) and all jobs of an order have to be delivered to the customer at the same time.

By Cheng and Wang (1999), even the problem $M2|d_j = d|\sum U_j$ is NP-hard, while the problem $Mm||\sum w_j U_j$ can be solved in $O(nP^m)$ time, where $P = \max_{1 \leq i \leq m} \sum_{1 \leq j \leq n} p_{i,j}$. Whether the complexity of the problem $M||\sum w_j U_j$ is strongly NP-hard is still open.

We show that, even when the weights are identical, the problem has no $(1-\varepsilon)\ln m$-approximation algorithm for any $\varepsilon > 0$ if NP is not a subset of DTIME($n^{\log\log n}$), and has no $c \cdot \ln m$-approximation algorithm for some constant $c > 0$ if P $\neq$ NP, where m is the number of machines. This also implies that the above problem is strongly NP-hard. Finally, we give a $(1+d)$-approximation algorithm for the problem

$$M|d_j = d, p_{i,j} = 0 \quad \text{or} \quad 1|\sum w_j U_j$$

2. IN-APPROXIMABILITY

To deduce the result of in-approximability, we use the following NP-hard minimum set cover problem (Garey and Johnson, 1979) for the reduction.

Minimum set cover. Given a collection $\mathcal{A} = \{A_1, A_2, \ldots, A_r\}$ of subsets of a finite set S with $\cup_{1 \leq i \leq r} A_i = S$, find a subset $\mathcal{A}' \subseteq \mathcal{A}$ with minimum cardinality such that every element in S belongs to at least one member of $\mathcal{A}'$. Where the subset $\mathcal{A}' \subseteq \mathcal{A}$ such that every element in S belongs to at least one member of $\mathcal{A}'$ is called a set cover of S on $\mathcal{A}$.

Theorem 2.1. Let S be the ground set in the minimum set cover problem. If there is a function $f(|S|)$ of $|S|$ such that the minimum set cover problem has no $f(|S|)$-approximation algorithm, then the scheduling problem

$$M|d_j = d, p_{i,j} = 0 \quad \text{or} \quad 1|\sum U_j$$

has no $f(m)$-approximation algorithm.

Proof. For a given instance of the minimum set cover problem, that is, a collection $\mathcal{A}=\{A_1, A_2, \ldots, A_r\}$ of a finite set S with $\cup_{1\le j\le r} A_j = S$, we construct an instance of the scheduling problem

$$M|d_j = d, p_{i,j} = 0 \quad \text{or} \quad 1|\sum U_j$$

as follows.

Let

$$t = \max_{x\in S} |\{A_j\colon x \in A_j\}|$$

$$t_x = t - |\{A_j\colon x \in A_j\}| \quad x \in S$$

We have $m=|S|$ machines with each element $x\in S$ corresponding to a machine M_x. We have

$$n = r + mt - \sum_{1\le j\le r} |A_j| = r + \sum_{x\in S} t_x$$

jobs as follows. Each set $A_j \in \mathcal{A}$ corresponds to a set-type job J_{A_j}, $1\le j\le r$, and each element $x\in S$ corresponds to t_x element-type jobs

$$J_{x(1)}, J_{x(2)}, \ldots, J_{x(t_x)}$$

Hence, we have r set-type jobs and $\sum_{x\in S} t_x$ element-type jobs. The set that consists of all jobs will be denoted by $\mathcal{J}$. The processing time $p_{x,y}$ of the operations $O_{x,y}$ ($x\in S$) of each job $J_y \in \mathcal{J}$ is defined as follows. For each set-type job J_{A_j},

$$p_{x,\,A_j} = \begin{cases} 1 & \text{if } x\in A_j \\ 0 & \text{otherwise} \end{cases}$$

For each element-type job $J_{z(i)}$, $z\in S$, $1\le i\le t_z$,

$$p_{x,\,z(i)} = \begin{cases} 1 & \text{if } x = z \\ 0 & \text{otherwise} \end{cases}$$

The common due date is defined as $d = t-1$.

Clearly, the construction can be done in polynomial time. We note the fact that

$$\sum_{J_y\in\mathcal{J}} p_{x,\,y} = t = d+1 \quad \text{for each machine } M_x$$

Hence, under any reasonable schedule, the completion time of every machine is $t=d+1$. On the other hand, because $\cup_{1\le j\le r} A_j = S$, for each machine M_x, there must be at least one set-type job J_{A_j} ($1\le j\le r$) such that $p_{x,A_j}=1$, that is, $x\in A_j$. We assume, without loss of generality, that every operation with processing time 0 is scheduled at time 0.

We first prove the following claim.

Claim 1. Let $\mathcal{A}^* \subseteq \mathcal{A}$ be a minimum set cover of S on $\mathcal{A}$ of the instance of the minimum set cover problem. Let π^* be an optimal schedule of the instance of our scheduling problem. Then,

$$|\mathcal{A}^*| = \sum_{J_y\in\mathcal{J}} U_y(\pi^*)$$

Proof. Let π be a schedule of the instance of our scheduling problem defined in such a way that all the element-type jobs and all the set-type jobs in $\{J_A: A \in \mathcal{A} \setminus \mathcal{A}^*\}$ are scheduled first, followed by the set-type jobs in $\{J_A: A \in \mathcal{A}^*\}$ arbitrarily. Because $\mathcal{A}^*$ is a set cover, for each machine M_x ($x \in S$), there must be a set-type job J_A with $A \in \mathcal{A}^*$ such that $x \in A$, and so $p_{x,A} = 1$. For the reason that the makespan of every machine is $t = d + 1$, all element-type jobs and all set-type jobs in $\{J_A: A \in \mathcal{A} \setminus \mathcal{A}^*\}$ are on time under the schedule π. This means that we have at most $|\mathcal{A}^*|$ tardy jobs under the schedule π. Hence, we have

$$|\mathcal{A}^*| \geq \sum_{J_y \in \mathcal{J}} U_y(\pi) \geq \sum_{J_y \in \mathcal{J}} U_y(\pi^*)$$

On the other hand, we consider the schedule π^*. If there is some element-type job $J_{x(i)}$ ($x \in S$, $1 \leq i \leq t_x$) that is tardy under π^*, that is, $C_{x,x(i)} = t = d + 1$, we can suppose that

$$X = \{x \in S\text{: there is } i \in [1, t_x] \text{ such that } C_{x,x(i)} = t = d + 1\}$$

For each $x \in X$, let $A(x) \in \mathcal{A}$ be such that $x \in A(x)$. By exchanging the positions of $O_{x,x(i)}$ and $O_{x,A(x)}$ on each machine M_x, $x \in X$, we get a schedule π' that is not worse than π^* and every element-type job is early under π'. Hence, we suppose, without loss of generality, that every element-type job is early under π^*. Let $\mathcal{J}^*$ be the set that consists of all the tardy jobs under the schedule π^*. Then $\mathcal{J}^*$ consists of set-type jobs only. Define $\mathcal{A}' = \{A \in \mathcal{A}: J_A \in \mathcal{J}^*\}$. Because the makespan of every machine is $t = d + 1$, for every element $x \in S$, there must be an operation $O_{x,A}$ ($A \in \mathcal{A}'$) that is tardy under π^*. Then, we have $C_{x,A} = t = d + 1$ and $p_{x,A} = 1$. This means that for every element $x \in S$, there must be a set $A \in \mathcal{A}'$ such that $x \in A$, and so $\mathcal{A}'$ is a set cover of S on $\mathcal{A}$. Hence, we have

$$\sum_{J_y \in \mathcal{J}} U_y(\pi^*) = |\mathcal{J}^*| = |\mathcal{A}'| \geq |\mathcal{A}^*|$$

This completes the proof of Claim 1. ∎

Now we suppose to the contrary that our scheduling problem has an $f(m)$-approximation algorithm ALG. Let π be the schedule obtained by the algorithm ALG. Let $\mathcal{A}^* \subseteq \mathcal{A}$ be a minimum set cover of S on $\mathcal{A}$ of the instance of the minimum set cover problem. Let π^* be an optimal schedule of the instance of our scheduling problem. Then, we have

$$\sum_{J_y \in \mathcal{J}} U_y(\pi) \leq f(m) \sum_{J_y \in \mathcal{J}} U_y(\pi^*)$$

By Claim 1, we also have

$$\sum_{J_y \in \mathcal{J}} U_y(\pi) \leq f(m)|\mathcal{A}^*|$$

Let $\mathcal{J}'$ be the set that consists of all the tardy jobs under the schedule π. We construct a set cover $\mathcal{A}'$ of S on $\mathcal{A}$ in the following way.

1. Set $\mathcal{A}' := \emptyset$, $X := S$.
2. If $X = \emptyset$, then stop; otherwise, go to 3.
3. Choose $x \in X$ arbitrarily. Let J_y be the job such that $C_{x,y}(\pi) = t = d + 1$. If J_y is a set-type job, then $y \in \mathcal{A}$ is a subset of S, and we set

$$\mathcal{A}' := \mathcal{A}' \cup \{y\} \quad \text{and} \quad X := X \setminus \{x\}$$

and then return to 2. If J_y is an element-type job, we arbitrarily choose a set $A \in \mathcal{A}$ such that $x \in A$, and we set

$$\mathcal{A}' := \mathcal{A}' \cup \{A\} \quad \text{and} \quad X := X \backslash \{x\}$$

and then return to 2.

It is easy to see that the set $\mathcal{A}'$ thus obtained is a set cover of S on $\mathcal{A}$ such that

$$|\mathcal{A}'| \leq \sum_{J_y \in \mathcal{J}} U_y(\pi)$$

Then, we have

$$|\mathcal{A}'| \leq f(m)|\mathcal{A}^*| = f(|S|)|\mathcal{A}^*|$$

This means that the minimum set cover problem has an $f(|S|)$-approximation algorithm, contradicting the assumption. The result follows.

By Feige (1998), for any given positive number ϵ, the minimum set cover problem has no $(1-\varepsilon)\ln|S|$-approximation algorithm, if NP is not a subset of DTIME($n^{\log\log n}$), where the definition of the notation DTIME($n^{\log\log n}$) can be found in Papadimitriou (1994). (It is conjectured in Papadimitriou (1994) that NP is not a subset of DTIME($n^{\log\log n}$).) By Raz and Safra (1997), there exists a constant $c > 0$, such that the minimum set cover problem has no $c \cdot \ln|S|$-approximation algorithm, if P $\neq$ NP. From Theorem 2.1, we deduce the following result.

Theorem 2.2. The scheduling problem

$$M|d_j = d, p_{i,j} = 0 \quad or \quad 1|\sum U_j$$

has no $(1-\varepsilon)\ln m$-approximation algorithm for any $\varepsilon > 0$ if NP is not a subset of DTIME($n^{\log\log n}$), and has no $c \cdot \ln m$-approximation algorithm for some constant $c > 0$ if $P \neq NP$.

The above result also implies that the scheduling problem $M|d_j = d$, $p_{i,j} = 0$ or $1|\sum U_j$ is strongly NP-hard. Since the minimum vertex cover problem (Garey, Johnson, and Stockmeyer, 1976) of graphs is a well-known special NP-hard subproblem of the minimum set cover problem, by using the decision version of the minimum vertex cover problem for the reduction, we can similarly deduce that the scheduling problem $M|d_j = 1$, $p_{i,j} = 0$ or $1|\sum U_j$ is also strongly NP-hard.

3. AN APPROXIMATION ALGORITHM

We now consider the weighted concurrent open shop scheduling problem

$$M|d_j = d, p_{i,j} = 0 \quad \text{or} \quad 1|\sum w_j U_j$$

Clearly a solution for this problem corresponds to a subset of jobs containing all tardy jobs. The objective is to find a subset of jobs $\mathcal{J}^* \subseteq \mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ such that

$$|\{J_j \in \mathcal{J} \backslash \mathcal{J}^*: p_{i,j} = 1\}| \leq d$$

for $1 \leq i \leq m$, and such that $\sum_{J_j \in \mathcal{J}^*} w_j$ is minimum.

For an instance I of the weighted scheduling problem, we set

$$P_i = \sum_{j=1}^{n} p_{i,j} \quad \text{for } i = 1, 2, \ldots, m$$

It is easy to see that the weighted concurrent open shop scheduling problem can be formulated as the following integer linear program ILP(I):

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{1\le j\le n} w_j x_j \\ \text{subject to} & \textstyle\sum_{1\le j\le n} p_{i,j} x_j \ge P_i - d \quad 1 \le i \le m \\ & x_j \in \{0,1\} \quad \text{for } j = 1, 2, \ldots, n \end{aligned}$$

Here, the job J_j with $x_j = 1$ will be tardy. By relaxing the integer constraints to simple nonnegativity constraints (i.e., $0 \le x_j \le 1$ for $j = 1, 2, \ldots, n$), we obtain the linear program LP(I):

$$\begin{aligned} \text{minimize} & \textstyle\sum_{1\le j\le n} w_j x_j \\ \text{subject to} & \textstyle\sum_{1\le j\le n} p_{i,j} x_j \ge P_i - d \ \ 1 \le i \le m \\ & 0 \le x_j \le 1 \quad \text{for } j = 1, 2, \ldots, n \end{aligned}$$

We give an approximation algorithm of the weighted scheduling problem as follows.

Algorithm 3.1. **LP-rounding Approximation**
Input: An instance I of the weighted concurrent open shop scheduling problem

$$M|d_j = d, p_{i,j} = 0 \quad \text{or} \quad 1|\sum w_j U_j$$

with m machines, n jobs $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$, a common due date d, n weights $w_1, w_2, \ldots, w_n$ and operation processing time $p_{i,j} = 0$ or 1.

Step 1: Set $P_i = \sum_{j=1}^{n} p_{i,j}$, for $i = 1, 2, \ldots, m$.
Step 2: Obtain the integer linear program formulation ILP(I) of the scheduling problem.
Step 3: Obtain the linear program LP(I) from ILP(I) by relaxing the integer constraints.
Step 4: Solve the linear program LP(I) and obtain an optimal solution $x^* = (x_1^*, x_2^*, \ldots, x_n^*)$ for LP(I).
Step 5: Obtain a subset of jobs $\mathcal{J}^*$ by setting

$$\mathcal{J}^* = \{J_j : x_j^* \ge 1/(d+1)\}$$

Output: The jobs in the subset $\mathcal{J}^*$ are tardy jobs.

Theorem 3.2. Algorithm 3.1 is a $(d+1)$*-approximation algorithm for the weighted concurrent open shop scheduling problem*

$$M|d_j = d, p_{i,j} = 0 \quad \text{or} \quad 1|\sum w_j U_j$$

Proof. Let $\mathcal{J}^*$ be the solution (i.e., a subset of jobs) obtained by Algorithm 3.1. If there is some i with $1 \le i \le m$ such that

$$|\{J_j \in \mathcal{J}^* : p_{i,j} = 1\}| \le P_i - d - 1$$

then by the definition of $\mathcal{J}^*$, we have

$$|\{J_j: p_{i,j} = 1, x_j^* \geq 1/(d+1)\}| \leq P_i - d - 1$$

Because

$$|\{J_j: p_{i,j} = 1\}| = P_i$$

it follows that

$$|\{J_j: p_{i,j} = 1, x_j^* < 1/(d+1)\}| \geq d + 1$$

Let

$$k = |\{J_j: p_{i,j} = 1, x_j^* < 1/(d+1)\}| \geq d + 1$$

Then, we have

$$\sum_{1 \leq j \leq n} p_{i,j} x_j^* < \frac{k}{d+1} + (P_i - k) = P_i - \frac{kd}{d+1} \leq P_i - d$$

This contradicts the fact that $\sum_{1 \leq j \leq n} p_{i,j} x_j^* \geq P_i - d$. Hence, we must have

$$|\{J_j \in \mathcal{J}^*: p_{i,j} = 1\}| \geq P_i - d \quad \text{for } 1 \leq i \leq m$$

This implies that $\mathcal{J}^*$ is a feasible solution of our scheduling problem.

Now let $V_{\text{opt}}(I)$ be the minimum weighted number of tardy jobs of our scheduling problem I, and let $V_{\text{LP}}^*(I)$ be the optimal measure of the relaxed linear program LP(I). Because $V_{\text{opt}}(I)$ is also the optimal measure of the integer linear program ILP(I), we clearly have

$$V_{\text{opt}}(I) \geq V_{\text{LP}}^*(I)$$

Since

$$\sum_{J_j \in \mathcal{J}^*} w_j \leq (d+1) \sum_{1 \leq j \leq n} w_j x_j^* = (d+1) V_{LP}^*(I) \leq (d+1) V_{\text{opt}}(I)$$

the result follows.

4. CONCLUSIONS

In this paper, we have studied the concurrent open shop scheduling problem to minimize the weighted number of tardy jobs with 0–1 operation processing times and a common due date d. We show that, even when the weights are identical, the problem has no $(1-\epsilon)\ln m$-approximation algorithm for any $\varepsilon > 0$ if NP is not a subset of DTIME($n^{\log\log n}$), and has no $c \cdot \ln m$-approximation algorithm for some constant $c > 0$ if $\text{P} \neq \text{NP}$, where m is the number of machines. This also implies that the problem is strongly NP-hard. We also give a $(1+d)$-approximation algorithm for the problem. When $d = 1$, this algorithm implies a well-known 2-approximation algorithm for the minimum vertex cover problem of graphs. To the best of our knowledge, for any given positive number δ with $0 < \delta < 1$, no $(1+\delta)$-approximation algorithm has been found for the minimum vertex cover problem of graphs. Hence, any improvement of our approximation algorithm is interesting.

ACKNOWLEDGEMENTS

This research was supported in part by the Research Grant Council of Hong Kong under grant number PolyU 5198/02E. We are thankful to an anonymous referee for his helpful comments on an earlier version of this paper.

REFERENCES

Cheng, T. C. E. and G. Wang, "Customer order scheduling on multiple facilities," Working Paper No. 11/98-9, Faculty of Business and Information Systems, The Hong Kong Polytechnic University, Hong Kong, 1999.

Feige, U., "A threshold of ln *n* for approximating set cover," *J. ACM*, **45**, 634–652 (1998).

Garey, M. R., D. S. Johnson, and L. Stockmeyer, "Some simplified NP-complete graph problem," *Theor. Comput. Sci.*, **1**, 237–267 (1976).

Garey, M. R., and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.

Papadimitriou, C. H., *Computational Complexity*, Addison-Wesley, NY, 1994.

Raz, R. and S. Safra, "A sub-constant error-probability low-degree test, and sub-constant error-probability PCP characterization of NP," *Proc. 29th ACM Symp. Theory Comput.*, ACM, 1997, pp. 475–484.