ELSEVIER

Operations Research Letters 21 (1997) 39–42

operations research letters

Polynomial algorithms for single machine scheduling problems with financial constraints

Jinxing Xie

Department of Applied Mathematics, Tsinghua University, Beijing 100084, People's Republic of China

Received 23 August 1995; revised 1 September 1996

Abstract

This paper deals with the single machine scheduling problem with multiple financial resource constraints. It is shown that the problem can be reduced to the two machine flow shop scheduling problem if the financial resources arrive uniformly over time, and it is also shown that the LPT (Largest Processing Time) rule generates an optimal solution to the problem if the financial resources are consumed uniformly by all the jobs. Hence there exist polynomial algorithms for these two special cases of the problem. © 1997 Elsevier Science B.V.

Keywords: Single machine scheduling; Financial constraints; Polynomial algorithms

1. Introduction

This paper deals with the single machine scheduling problem with multiple financial resource constraints. A financial resource is a typical non-renewable resource which is arriving dynamically over time and will be actually consumed by the jobs competing for it. There are k different types of financial resources which are required to process n jobs. In each time period t an amount of α_{ht} ($h = 1,2, \ldots, k$; $t = 1,2, \ldots$) units of resource h is arriving. To process job j an amount of r_{jh} ($j = 1,2, \ldots, n$; $h = 1,2, \ldots, k$) units of resource h must be available at its starting time and these amounts of resources are completely consumed by job j. The problem is to find a schedule to minimize the makespan.

In case of arbitrary resource requirements and availability, Carlier [1] proved that the non-preemptive single machine scheduling problem with financial constraints is NP-complete if job processing times are different from unity. When money is the only resource and the jobs are precedence-related, Carlier and Rinnooy Kan [2] developed a polynomial algorithm. For the preemptive parallel machine scheduling problems with financial constraints, Slowinski [4] presented a polynomial algorithm. Under the assumption that there exists only one financial resource and it arrives uniformly (i.e. the arriving amount of this resource is constant over time periods), Toker et al. [5] proved that the single machine scheduling problem can be reduced to the two machine flow shop scheduling problem [3]. Using FC_k (k is omitted when $k = 1$) to stand for that there are k different types of

PII S0167-6377(97)00007-2

financial resources, the result of Toker et al. [5] can be rewritten as follows:

The scheduling problem 1/FC: $\alpha_t = 1/C_{\max}$ *can be reduced to the two machine flow shop scheduling problem* $F2//C_{\max}$.

This paper generalizes this result to the problem with multiple financial resource constraints. It is shown that the problem can be reduced to the two machine flow shop scheduling problem if the financial resources arrive uniformly over time, and it is also shown that the LPT (Largest Processing Time) rule is optimal if the financial resources are consumed uniformly by all the jobs. Hence there exist polynomial algorithms for these two special cases of the problem.

2. Polynomial algorithms

Theorem 1. *The scheduling problem* $1/FC_k$: $\alpha_{ht} = 1/C_{\max}$ *can be reduced to k two machine flow shop scheduling problems* $F2//C_{\max}$.

Proof. Consider the scheduling problem $1/FC_k$: $\alpha_{ht} = 1/C_{\max}$. The n jobs are labeled as J_j $(j = 1,2, \dots, n)$. Let p_j be the processing time of J_j and r_{jh} the amount of financial resource $h\,(1 \leqslant h \leqslant k)$ required to process job J_j. For a given schedule, let $J_{[j]}$ be the job at the jth position. Then $r_{[j]h}$ is the amount of financial resource $h\,(1 \leqslant h \leqslant k)$ required to process job $J_{[j]}$ and $p_{[j]}$ is the processing time of job $J_{[j]}$. Let $I_{[j]}$ be the idle time on the machine before job $J_{[j]}$ is processed. Since one unit of each resource becomes available each period,

$$I_{[1]} = \max_{1 \leqslant h \leqslant k} \{r_{[1]h}\}, \tag{1}$$

and

$$I_{[j]} = \max\left\{0, \max_{1 \leqslant h \leqslant k} \left\{\sum_{i=1}^{j} r_{[i]h}\right\} - \sum_{i=1}^{j-1} p_{[i]} - \sum_{i=1}^{j-1} I_{[i]}\right\} \quad (j = 2,3, \dots, n). \tag{2}$$

From (2), we have

$$\begin{aligned}\sum_{i=1}^{j} I_{[i]} &= \sum_{i=1}^{j-1} I_{[i]} + I_{[j]} \\ &= \sum_{i=1}^{j-1} I_{[i]} + \max\left\{0, \max_{1 \leqslant h \leqslant k} \left\{\sum_{i=1}^{j} r_{[i]h}\right\} - \sum_{i=1}^{j-1} p_{[i]} - \sum_{i=1}^{j-1} I_{[i]}\right\} \\ &= \max\left\{\sum_{i=1}^{j-1} I_{[i]}, \max_{1 \leqslant h \leqslant k} \left\{\sum_{i=1}^{j} r_{[i]h}\right\} - \sum_{i=1}^{j-1} p_{[i]}\right\} \quad (j = 2,3, \dots, n).\end{aligned} \tag{3}$$

From (1) and (3), the total idle time becomes

$$\begin{aligned}\sum_{i=1}^{n} I_{[i]} &= \max\left\{I_{[1]}, \max_{2 \leqslant j \leqslant n} \max_{1 \leqslant h \leqslant k} \left\{\sum_{i=1}^{j} r_{[i]h} - \sum_{i=1}^{j-1} p_{[i]}\right\}\right\} \\ &= \max_{1 \leqslant h \leqslant k} \left\{\max_{2 \leqslant j \leqslant n} \left\{r_{[1]h}, \sum_{i=1}^{j} r_{[i]h} - \sum_{i=1}^{j-1} p_{[i]}\right\}\right\}.\end{aligned} \tag{4}$$

Hence the makespan is

$$\begin{aligned}C_{\max} &= \sum_{i=1}^{n} p_{[i]} + \sum_{i=1}^{n} I_{[i]} \\ &= \sum_{i=1}^{n} p_{[i]} + \max_{1 \leqslant h \leqslant k} \left\{\max_{2 \leqslant j \leqslant n} \left\{r_{[1]h}, \sum_{i=1}^{j} r_{[i]h} - \sum_{i=1}^{j-1} p_{[i]}\right\}\right\} \\ &= \max_{1 \leqslant h \leqslant k} \left\{\sum_{i=1}^{n} p_{[i]} + \max_{2 \leqslant j \leqslant n} \left\{r_{[1]h}, \sum_{i=1}^{j} r_{[i]h} - \sum_{i=1}^{j-1} p_{[i]}\right\}\right\} \\ &= \max_{1 \leqslant h \leqslant k} C_{\max}^{h}\end{aligned} \tag{5}$$

where

$$C_{\max}^{h} = \sum_{i=1}^{n} p_{[i]} + \max_{2 \leqslant j \leqslant n} \left\{ r_{[1]h}, \sum_{i=1}^{j} r_{[i]h} - \sum_{i=1}^{j-1} p_{[i]} \right\} \quad (1 \leqslant h \leqslant k). \tag{6}$$

Now consider the scheduling problem $F2//C_{\max}$. Let p_{j1} and p_{j2} be the processing times of job J_j on the first machine and the second machine, respectively. Clearly there is no idle time between the jobs on the first machine. For a given schedule, let $J_{[j]}$ be the job at the jth position and $I'_{[j]}$ be the idle time on the second machine before job $J_{[j]}$ is processed. Then

$$I'_{[1]} = p_{[1]1}, \tag{7}$$

and

$$I'_{[j]} = \max\left\{0, \sum_{i=1}^{j} p_{[i]1} - \sum_{i=1}^{j-1} p_{[i]2} - \sum_{i=1}^{j-1} I'_{[i]}\right\} \quad (j = 2, 3, \ldots, n). \tag{8}$$

From (7) and (8), we have

$$\sum_{i=1}^{n} I'_{[i]} = \max_{2 \leqslant j \leqslant n} \left\{ p_{[1]1}, \sum_{i=1}^{j} p_{[i]1} - \sum_{i=1}^{j-1} p_{[i]2} \right\}. \tag{9}$$

Hence the makespan is

$$C'_{\max} = \sum_{i=1}^{n} p_{[i]2} + \max_{2 \leqslant j \leqslant n} \left\{ p_{[1]1}, \sum_{i=1}^{j} p_{[i]1} - \sum_{i=1}^{j-1} p_{[i]2} \right\}. \tag{10}$$

In comparison of Eq. (10) with (6), taking r_{ih} and p_i of the scheduling problem $1/FC_k$: $\alpha_{ht} = 1/C_{\max}$ as p_{i1} and p_{i2} of the problem $F2//C_{\max}$, respectively, it is known that Eqs. (10) and (6) are identical for any given h $(1 \leqslant h \leqslant k)$. This completes the proof.

Corollary 1. *The problem* $1/FC_k$: $\alpha_{ht} = 1/C_{\max}$ *can be solved in* $O(kn \log n)$ *time.*

Proof. Since the two machine flow shop scheduling problem $F2//C_{\max}$ can be solved in $O(n \log n)$ time [3], Corollary 1 is true according to Theorem 1.

Theorem 2. *The LPT (Largest Processing Time) rule generates an optimal solution to the scheduling problem* $1/FC_k$: $r_{jh} = r_h/C_{\max}$.

Proof. Let the n jobs be labeled as J_j $(j = 1,2, \ldots, n)$ and p_j be the processing time of J_j. For a given schedule, let $J_{[j]}$ be the job at the jth position and t_j $(j = 1,2, \ldots, n)$ be the earliest time possible to process the job $J_{[j]}$. Then

$$t_j = \max_{1 \leqslant h \leqslant k} \min\left\{ t: \sum_{\tau=1}^{t} \alpha_{h\tau} \geqslant j \times r_h \right\} \quad (j = 1,2, \ldots, n). \tag{11}$$

Let $I_{[j]}$ be the idle time on the machine before $J_{[j]}$ is processed. Then

$$I_{[1]} = t_1, \tag{12}$$

and

$$I_{[j]} = \max\left\{0, t_j - \sum_{i=1}^{j-1} p_{[i]} - \sum_{i=1}^{j-1} I_{[i]}\right\} \quad (j = 2,3, \ldots, n). \tag{13}$$

From (12) and (13), the total idle time is

$$\sum_{i=1}^{n} I_{[i]} = \max\left\{ t_1, \max_{2 \leqslant j \leqslant n} \left\{ t_j - \sum_{i=1}^{j-1} p_{[i]} \right\} \right\}. \tag{14}$$

Hence the makespan is

$$C_{\max} = \sum_{i=1}^{n} p_{[i]} + \sum_{i=1}^{n} I_{[i]} = \sum_{i=1}^{n} p_{[i]} + \max_{2 \leqslant j \leqslant n} \left\{ t_1, t_j - \sum_{i=1}^{j-1} p_{[i]} \right\}. \tag{15}$$

Noticing that t_j $(j = 1, 2, \ldots, n)$ is independent on the given schedule, the LPT rule generates an optimal solution to the problem. This completes the proof. □

Corollary 2. *The problem* $1/FC_k$: $r_{jh} = r_h/C_{\max}$ *can be solved in* $O(n \log n)$ *time.*

Proof. Since the problem of sorting n numbers can be solved in $O(n \log n)$ time, Corollary 2 is true according to Theorem 2.

Acknowledgements

This work has been partially supported by the National Natural Science Foundation of China (Project No. 69584005). The author would like to express his indebtedness to the anonymous referees for their helpful comments and suggestions.

References

[1] J. Carlier, "Complexité des problèmes d'ordonnancement à constraintes de financement", in: P. Hansen, D. de Werra (eds.), *Regards sur la Théorie des Graphes*, Presses Polytechniques Romandes, Lausanne, 1980, pp. 183–186.

[2] J. Carlier and A.H.G. Rinnooy Kan, "Scheduling subject to nonrenewable-resource constraints", *Oper. Res. Lett.* **1**, 52–55 (1982).

[3] S.M. Johnson, "Optimal two and three stage production schedules with setup time included", *Naval Res. Logist. Quart.* **1**, 61–68 (1954).

[4] R. Slowinski, "Preemptive scheduling of independence jobs on parallel machine subject to financial constraints", *European J. Oper. Res.* **15**, 366–373 (1984).

[5] A. Toker, S. Kondakci and N. Erkip, "Scheduling under a non-renewable resource constraint", *J. Oper. Res. Soc.* **42**, 811–814 (1991).