An FPTAS for parallel-machine scheduling under a grade of service provision to minimize makespan

Min Ji[1]
College of Computer Science & Information Engineering,
Zhejiang Gongshang University, Hangzhou 310035, P.R. China

T. C. E. Cheng[2]
Department of Logistics, The Hong Kong Polytechnic University,
Kowloon, Hong Kong

---

[1] Email: jimkeen@163.com

[2] Corresponding author. Email: LGTcheng@polyu.edu.hk

Abstract

We consider the m parallel-machine scheduling problem that process service requests from various customers who are entitled to different levels of grade of service (GoS). The objective is to minimize the makespan. We give a fully polynomial-time approximation scheme for the case where m is fixed.

Keywords. Machine scheduling; Eligibility; Grade of service; Makespan

1 Introduction

It is common practice in any service industry to provide differential services to customers based on their entitled privileges assigned according to their promised levels of grade of service (GoS). Jobs are allowed to be processed on a particular machine when the GoS level of the job is no less than the GoS level of the machine. In fact, the processing capability of the machines labelled with a high GoS level tends to be reserved for jobs with a high GoS level. Hence, if we assign relatively high GoS levels to the jobs from valued customers, we can ensure providing better service to more valued customers. In such situations, assigning jobs to machines becomes a parallel-machine scheduling problem with a special eligibility constraint.

The problem under consideration can be formally described as follows: There are n independent jobs $\mathcal{J} = \{J_1, J_2, \cdots, J_n\}$ and m identical machines $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$. The processing time of job J_j is p_j. Each job J_j and each machine M_i are labelled with the GoS levels $G(J_j)$ and $G(M_i)$, respectively. Job J_j is allowed to be processed on machine M_i only when $G(M_i) \leq G(J_j)$. A feasible schedule is, then, a partition of $\mathcal{J}$ into m disjoint sets, $\mathcal{S} =< S_1, S_2, \cdots, S_m >$ such that S_i is allowed to include J_j only if $G(M_i) \leq G(J_j)$. Let C_j be the completion time of job J_j in a schedule. The objective is to minimize the makespan, i.e., $C_{\max} = \max_{j=1,2,\cdots,n} C_j$. Using the three-field notation of Graham et al. [1], we denote this scheduling model as $P_m|GoS|C_{\max}$.

The above defined problem dates back to Hwang et al. [2]. They proposed an approximation algorithm LG-LPT, and proved that its makespan is not greater than $\frac{5}{4}$ times the optimal makespan for $m = 2$ and not greater than $2 - \frac{1}{m-1}$ times the optimal makespan for $m \geq 3$. Jiang [5], Park et al. [9] and Jiang et al. [6] investigated the semi-online and online versions of the considered model.

In this paper we present a fully polynomial-time approximation scheme (FPTAS) for the problem $P_m|GoS|C_{\max}$ with a fixed number of m, which greatly improves the bound in Hwang et al. [2]. The design of the FPTAS closely follows our earlier works [3, 4], in which two FPTASs were presented for two time-dependent scheduling problems where each job can be processed on any machine. Consequently, the basic descriptions in this paper are similar to those in [3, 4]. Specifically, we design a modified FPTAS in this paper to deal with the special eligibility constraint that the jobs and machines each have a GoS level. So the modified FPTAS makes a contribution to the practice of scheduling.

The presentation of this paper is organized as follows. In Section 2 we propose an FPTAS for the problem $P_m|GoS|C_{\max}$, where m is fixed, and prove its correctness and establish its time complexity. We conclude the paper in Section 3.

2 An FPTAS

An algorithm $\mathcal{A}$ is called a $(1+\varepsilon)$-*approximation* algorithm for a minimization problem if it produces a solution that is at most $1+\varepsilon$ times as big as the optimal value, running in time that is polynomial in the input size of the problem instance. A family of approximation algorithms $\{\mathcal{A}_\varepsilon\}$ is called a *fully polynomial-time approximation scheme* (FPTAS) if, for each $\varepsilon > 0$, the algorithm $\mathcal{A}_\varepsilon$ is a $(1+\varepsilon)$-approximation algorithm that is polynomial in the input size of the problem instance and in $1/\varepsilon$. From now on we assume, without loss of generality, that $0 < \varepsilon \leq 1$. If $\varepsilon > 1$, then a 2-approximation algorithm can be taken as a $(1+\varepsilon)$-approximation algorithm.

Without loss of generality, we assume that all the machines are indexed in nondecreasing order of $G(M_i)$ so that $G(M_1) \leq G(M_2) \leq \cdots \leq G(M_m)$. We first define $s_j = max\{i|G(M_i) \leq G(J_j)\}$ for each $j = 1, 2, \cdots, n$. Therefore job J_j can be processed on machine $M_1, \cdots, M_{s_j}$. Then we introduce variables x_j, $j = 1, 2, \cdots, s_j$, where $x_j = k$ if job J_j is processed on machine M_k, $k \in \{1, 2, \cdots, s_j\}$. Let X be the set of all the vectors $x = (x_1, x_2, \cdots, x_n)$ with $x_j = k$, $j = 1, 2, \cdots, n$, $k = 1, 2, \cdots, s_j$. We define the following initial and recursive functions on X:

$$\begin{aligned} f_0^i(x) &= 0, \ i = 1, 2, \cdots, m, \\ f_j^k(x) &= f_{j-1}^k(x) + p_j, \ \text{for } x_j = k, \\ f_j^i(x) &= f_{j-1}^i(x), \ \text{for } x_j = k, i \neq k, \end{aligned}$$

Thus, the problem $P_m|GoS|C_{\max}$ reduces to the following problem:

$$\text{Minimize } Q(x) \text{ for } x \in X, \text{ where } Q(x) = \max_{i=1,2,\cdots,m} f_n^i(x).$$

We first introduce the procedure $Partition(A, e, \delta)$ proposed by Kovalyov and Kubiak [7, 8], where $A \subseteq X$, e is a nonnegative integer function on X, and $0 < \delta \leq 1$. This procedure partitions A

into disjoint subsets $A^e_1, A^e_2, \cdots, A^e_{k_e}$ such that $|e(x)-e(x')| \le \delta \min\{e(x), e(x')\}$ for any x, x' from the same subset A^e_j, $j = 1, 2, \cdots, k_e$. The following description provides the details of $Partition(A, e, \delta)$.

Procedure $Partition(A, e, \delta)$

Step 1. Arrange vectors $x \in A$ in the order $x^{(1)}, x^{(2)}, \cdots, x^{(|A|)}$ such that $0 \le e(x^{(1)}) \le e(x^{(2)}) \le \cdots \le e(x^{(|A|)})$.

Step 2. Assign vectors $x^{(1)}, x^{(2)}, \cdots, x^{(i_1)}$ to set A^e_1 until i_1 is found such that $e(x^{(i_1)}) \le (1+\delta)e(x^{(1)})$ and $e(x^{(i_1+1)}) > (1+\delta)e(x^{(1)})$. If such i_1 does not exist, then take $A^e_{k_e} = A^e_1 = A$, and stop. Assign vectors $x^{(i_1+1)}, x^{(i_1+2)}, \cdots, x^{(i_2)}$ to set A^e_2 until i_2 is found such that $e(x^{(i_2)}) \le (1+\delta)e(x^{(i_1+1)})$ and $e(x^{(i_2+1)}) > (1+\delta)e(x^{(i_1+1)})$. If such i_2 does not exist, then take $A^e_{k_e} = A^e_2 = A - A^e_1$, and stop. Continue the above construction until $x^{(|A|)}$ is included in $A^e_{k_e}$ for some k_e. ■

Procedure $Partition$ requires $O(|A| \log |A|)$ operations to arrange the vectors of A in nondecreasing order of $e(x)$, and $O(|A|)$ operations to provide a partition. The main properties of $Partition$ that will be used in the development of our FPTAS $\{\mathcal{A}^m_\varepsilon\}$ were presented in Kovalyov and Kubiak [7, 8] as follows.

Property 1 $|e(x) - e(x')| \le \delta \min\{e(x), e(x')\}$ *for any* $x, x' \in A^e_j$, $j = 1, 2, \cdots, k_e$.

Property 2 $k_e \le \log e(x^{(|A|)})/\delta + 2$ *for* $0 < \delta \le 1$ *and* $1 \le e(x^{(|A|)})$.

A formal description of the FPTAS $\mathcal{A}^m_\varepsilon$ for the problem $P_m|GoS|C_{\max}$ is given below.

Algorithm $\mathcal{A}^m_\varepsilon$

Step 1. (Initialization) Number the machines in nondecreasing order of $G(M_i)$ so that $G(M_1) \le G(M_2) \le \cdots \le G(M_m)$. Set $Y_0 = \{(0, 0, \cdots, 0)\}$ and $j = 1$.

Step 2. (Generation of $Y_1, Y_2, \cdots, Y_n$) For set Y_{j-1}, generate Y'_j by adding k, $k = 1, 2, \cdots, s_j$, in position j of each vector from Y_{j-1}. Calculate the following for any $x \in Y'_j$, assuming $x_j = k$:

$$\begin{aligned} f^k_j(x) &= f^k_{j-1}(x) + p_j, \\ f^i_j(x) &= f^i_{j-1}(x), \text{ for } i \ne k, \end{aligned}$$

If $j = n$, then set $Y_n = Y'_n$, and go to Step 3.

If $j < n$, then set $\delta = \varepsilon/(2(n+1))$, and perform the following computations.

Call $Partition(Y'_j, f^i_j, \delta)$ to partition set Y'_j into disjoint subsets $Y^{fi}_1, Y^{fi}_2, \cdots, Y^{fi}_{k_{fi}}$ $(i = 1, 2, \cdots, m)$.

Divide set Y'_j into disjoint subsets $Y_{a_1\cdots a_m} = Y^{f^1}_{a_1} \cap \cdots \cap Y^{f^m}_{a_m}$, $a_1 = 1, 2, \cdots, k_{f^1}$; $\cdots$; $a_m = 1, 2, \cdots, k_{f^m}$. For each nonempty subset $Y_{a_1\cdots a_m}$, choose a vector $x^{(a_1\cdots a_m)}$ such that

$$f^i_j(x^{(a_1\cdots a_m)}) = \min\{\max_{i=1,2,\cdots,m} f^i_j(x) \mid x \in Y_{a_1\cdots a_m}\}.$$

Set $Y_j := \{x^{(a_1\cdots a_m)} \mid a_1 = 1, 2, \cdots, k_{f^1}; \cdots; a_m = 1, 2, \cdots, k_{f^m}$ and $Y^{f^1}_{a_1} \cap \cdots \cap Y^{f^m}_{a_m} \neq \emptyset\}$, and $j = j + 1$.

Repeat Step 2.

Step 3. (Solution) Select vector $x^0 \in Y_n$ such that $Q(x^0) = \min\{Q(x) \mid x \in Y_n\} = \min\{\max_{i=1,2,\cdots,m} f^i_n(x) \mid x \in Y_n\}$. ■

Let $x^* = (x^*_1, x^*_2, \cdots, x^*_n)$ be an optimal solution for the problem $P_m|GoS|C_{\max}$ with a fixed number of machines. Let $L = \log(\max\{n, 1/\varepsilon, p_{\max}\})$, where $p_{\max} = \max_{j=1,2,\cdots,n} p_j$. We show the main result of this section in the following.

Theorem 1 *Algorithm $\mathcal{A}^m_\varepsilon$ finds $x^0 \in X$ for the problem $P_m|GoS|C_{\max}$ such that $Q(x^0) \leq (1 + \varepsilon)Q(x^*)$ in $O(n^{m+1}L^{m+1}/\varepsilon^m)$.*

Proof. Suppose that $(x^*_1, \cdots, x^*_j, 0, \cdots, 0) \in Y_{a_1\cdots a_m} \subseteq Y'_j$ for some j and $a_1, \cdots a_m$. By the definition of $\mathcal{A}^m_\varepsilon$, such a j always exists (e.g., $j = 1$). Algorithm $\mathcal{A}^m_\varepsilon$ may not choose $(x^*_1, \cdots, x^*_j, 0, \cdots, 0)$ for further construction; however, for a vector $x^{(a_1\cdots a_m)}$ chosen instead of it, we have

$$|f^i_j(x^*) - f^i_j(x^{(a_1\cdots a_m)})| \leq \delta f^i_j(x^*), i = 1, \cdots, m,$$

due to Property 1. Set $\delta_1 = \delta$. We consider vector $(x^*_1, \cdots, x^*_j, x^*_{j+1}, 0, \cdots, 0)$ and $\tilde{x}^{(a_1\cdots a_m)} = (x^{(a_1\cdots a_m)}_1, \cdots, x^{(a_1\cdots a_m)}_j, x^*_{j+1}, 0, \cdots, 0)$. Without loss of generality, we assume $x^*_{j+1} = k$. It follows that

$$|f^k_{j+1}(x^*) - f^k_{j+1}(\tilde{x}^{(a_1\cdots a_m)})|$$

$$\begin{aligned} &= |(f_j^k(x^*) + p_{j+1}) - (f_j^k(x^{(a_1\cdots a_m)}) + p_{j+1})| \\ &= |(f_j^k(x^*) - f_j^k(x^{(a_1\cdots a_m)}))| \\ &\le \delta_1 f_j^k(x^*) \le \delta_1 f_{j+1}^k(x^*), \end{aligned} \tag{1}$$

Consequently,

$$f_{j+1}^k(\tilde{x}^{(a_1\cdots a_m)}) \le (1+\delta_1) f_{j+1}^k(x^*).$$

Similarly, for $i \neq k$, we have

$$|f_{j+1}^i(x^*) - f_{j+1}^i(\tilde{x}^{(a_1\cdots a_m)})| \le \delta_1 f_{j+1}^i(x^*), \tag{2}$$

and

$$f_{j+1}^i(\tilde{x}^{(a_1\cdots a_m)}) \le (1+\delta_1) f_{j+1}^i(x^*).$$

Assume that $\tilde{x}^{(a_1\cdots a_m)} \in Y_{c_1\cdots c_m} \subseteq Y'_{j+1}$ and Algorithm $\mathcal{A}_\varepsilon^m$ chooses $x^{(c_1\cdots c_m)} \in Y_{c_1\cdots c_m}$ instead of $\tilde{x}^{(a_1\cdots a_m)}$ in the $(j+1)$-th iteration. We have

$$|f_{j+1}^i(\tilde{x}^{(a_1\cdots a_m)}) - f_{j+1}^i(x^{(c_1\cdots c_m)})| \le \delta f_{j+1}^i(\tilde{x}^{(a_1\cdots a_m)}) \le \delta(1+\delta_1) f_{j+1}^i(x^*), i = 1, \cdots, m. \tag{3}$$

For $i = 1, 2, \cdots, m$, from (1), (2) and (3), we obtain

$$\begin{aligned} & |f_{j+1}^i(x^*) - f_{j+1}^i(x^{(c_1\cdots c_m)})| \\ \le\ & |f_{j+1}^i(x^*) - f_{j+1}^i(\tilde{x}^{(a_1\cdots a_m)})| + |f_{j+1}^i(\tilde{x}^{(a_1\cdots a_m)}) - f_{j+1}^i(x^{(c_1\cdots c_m)})| \\ \le\ & (\delta_1 + \delta(1+\delta_1)) f_{j+1}^i(x^*) \\ =\ & (\delta + \delta_1(1+\delta)) f_{j+1}^i(x^*). \end{aligned} \tag{4}$$

Set $\delta_l = \delta + \delta_{l-1}(1+\delta)$, $l = 2, 3, \cdots, n-j+1$. From (4), we obtain

$$|f_{j+1}^i(x^*) - f_{j+1}^i(x^{(c_1\cdots c_m)})| \le \delta_2 f_{j+1}^i(x^*).$$

Repeating the above argument for $j+2, \cdots, n$, we show that there exists $x' \in Y_n$ such that

$$|f_n^i(x^*) - f_n^i(x')| \le \delta_{n-j+1} f_n^i(x^*), i = 1, 2, \cdots, m.$$

Since

$$\begin{aligned}
\delta_{n-j+1} &\leq \delta\sum_{j=0}^{n}(1+\delta)^j \\
&= (1+\delta)^{n+1}-1 \\
&= \sum_{j=1}^{n+1}\frac{(n+1)n\cdots(n-j+2)}{j!}\delta^j \\
&= \sum_{j=1}^{n+1}\frac{(n+1)n\cdots(n-j+2)}{j!(n+1)^j}(\frac{\varepsilon}{2})^j \\
&\leq \sum_{j=1}^{n+1}\frac{1}{j!}(\frac{\varepsilon}{2})^j \\
&\leq \sum_{j=1}^{n+1}(\frac{\varepsilon}{2})^j \\
&\leq \varepsilon\sum_{j=1}^{n+1}(\frac{1}{2})^j \\
&\leq \varepsilon.
\end{aligned}$$

Therefore, we have

$$|f_n^i(x^*)-f_n^i(x')|\leq\varepsilon f_n^i(x^*), i=1,2,\cdots,m.$$

It implies

$$|\max_{i=1,2,\cdots,m}f_n^i(x')-\max_{i=1,2,\cdots,m}f_n^i(x^*)|\leq\varepsilon\max_{i=1,2,\cdots,m}f_n^i(x^*).$$

Then in Step 3, vector x^0 will be chosen such that

$$\begin{aligned}
&|\max_{i=1,2,\cdots,m}f_n^i(x^0)-\max_{i=1,2,\cdots,m}f_n^i(x^*)| \\
\leq\ &|\max_{i=1,2,\cdots,m}f_n^i(x')-\max_{i=1,2,\cdots,m}f_n^i(x^*)| \\
\leq\ &\varepsilon\max_{i=1,2,\cdots,m}f_n^i(x^*).
\end{aligned}$$

Therefore we have $Q(x^0)\leq(1+\varepsilon)Q(x^*)$.

The time complexity of Algorithm $\mathcal{A}_\varepsilon^m$ can be established by noting that the most time-consuming operation of iteration j of Step 2 is a call of procedure *Partition*, which requires $O(|Y_j'|\log|Y_j'|)$ time to complete. To estimate $|Y_j'|$, recall that $|Y_{j+1}'|\leq m|Y_j|\leq mk_f^1k_f^2\cdots k_f^m$. By Property 2, we have $k_f^i\leq 2(n+1)\log(np_{\max})/\varepsilon+2\leq 2(n+1)L/\varepsilon+2$, $i=1,2,\cdots,m$. Thus, $|Y_j'|=$

$O(n^m L^m/\varepsilon^m)$, and $|Y'_j| \log |Y'_j| = O(n^m L^{m+1}/\varepsilon^m)$. Therefore, the time complexity of Algorithm $\mathcal{A}^m_\varepsilon$ is $O(n^{m+1}L^{m+1}/\varepsilon^m)$. ■

3 Conclusion

This paper studied the m parallel-machine scheduling problem under a grade of service provision. For the objective of minimizing makespan, we gave a fully polynomial-time approximation scheme for the case where m is fixed. Future research may focus on other scheduling objectives.

Acknowledgment

We thank two anonymous referees for their helpful comments on an earlier version of our paper. Ji was supported by the National Natural Science Foundation of China under grant number 60673179, and Zhejiang Gongshang University under grant numbers 1130XJ030621 and Q07-03. Cheng was supported in part by The Hong Kong Polytechnic University under a grant from the *Area of Strategic Development in China Business Services.*

References

[1] R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey, *Annals of Discrete Mathematics*, 5 (1979) 287-326.

[2] H.C. Hwang, S.Y. Chang, K. Lee, Parallel machine scheduling under a grade of service provision, *Computers & Operations Research*, 31 (2004) 2055-2061

[3] M. Ji and T.C.E. Cheng, An FPTAS for scheduling jobs with piecewise linear decreasing processing times to minimize makespan, *Information Processing Letters*, 102 (2007) 41-47.

[4] M. Ji and T.C.E. Cheng, Parallel-machine scheduling with simple linear deterioration to minimize total completion time, *European Journal of Operational Research*, 188 (2008) 342-347.

[5] Y.W. Jiang, Online scheduling on parallel machines with two GoS levels, *Lecture Notes in Computer Science*, 4041 (2006) 11-21.

[6] Y.W. Jiang, Y. He, C.M. Tang, Optimal online algorithms for scheduling on two identical machines under a grade of service, *Journal of Zhejiang University - Science A*, 7 (2006) 309-314.

[7] M.Y. Kovalyov, W. Kubiak, A fully polynomial approximation scheme for minimizing makespan of deteriorating jobs, *Journal of Heuristics*, 3 (1998) 287-297.

[8] M.Y. Kovalyov, W. Kubiak, A fully polynomial approximation scheme for the weighted earliness-tardiness problem, *Operations Research*, 47 (1999) 757-761.

[9] J. Park, S.Y. Chang, K. Lee, Online and semi-online scheduling of two machines under a grade of service provision, *Operations Research Letters*, 34 (2006) 692-696.