ANZIAM J. 58(2017), 306–313
doi:10.1017/S1446181117000190

PARALLEL MACHINE SCHEDULING WITH JOB DELIVERY COORDINATION

J. M. DONG[1], X. S. WANG[2], L. L. WANG[3] and J. L. HU[✉1]

(Received 30 June, 2016; accepted 15 October, 2016; first published online 25 May 2017)

Abstract

We analyse a parallel (identical) machine scheduling problem with job delivery to a single customer. For this problem, each job needs to be processed on m parallel machines non-pre-emptively and then transported to a customer by one vehicle with a limited physical capacity. The optimization goal is to minimize the makespan, the time at which all the jobs are processed and delivered and the vehicle returns to the machine. We present an approximation algorithm with a tight worst-case performance ratio of $7/3 - 1/m$ for the general case, $m \geq 3$.

2010 *Mathematics subject classification*: primary 90B35; secondary 90C27.

Keywords and phrases: scheduling, job delivery, bin packing, approximation algorithm.

1. Introduction

Coordinated production–transportation scheduling problems have received much attention from both industry and academic researchers. In this paper, we consider the scheduling problem in a parallel machine environment, in which jobs finished on the parallel machines need to be delivered to a customer by a single vehicle with a limited load capacity. And, the vehicle has to return to the machine for another shipment after it delivers a shipment to the customer. The goal is to minimize the makespan, that is, the time at which all the jobs are processed and delivered to the customer and the vehicle returns to the machine.

Our target scheduling problem is formally described as follows. We are given m parallel (identical) machines and a set of jobs $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ to be processed on the machines. Each job J_j has a non-pre-emptive processing time p_j on the machine

[1]School of Sciences, Zhejiang Sci-Tech University, Hangzhou 310018, China;
e-mail: djm226@163.com, hujlhz@163.com.
[2]Department of Commodity, Yongan Futures Co. Ltd, Hangzhou 310000, China;
e-mail: xueshi1989@126.com.
[3]College of Automation, Hangzhou Dianzi University, Hangzhou 310018, China;
e-mail: liangzi@hdu.edu.cn.

and a physical size s_j. The finished jobs are to be delivered to a customer by a single vehicle and they can be transported by the vehicle in batches. The size s_j of the job J_j represents its fractional space requirement on the vehicle and a batch is *valid* only if the total size of the jobs therein does not exceed 1. The vehicle takes a constant time T for a round trip between the machine and the customer. The objective is to minimize the makespan $C_{\max}$. This problem can be denoted as $(Pm \to D, k = 1 \mid v = 1, c = 1 \mid C_{\max})$, where "$Pm \to D, k = 1$" states that the jobs are first processed on m parallel machines and then delivered to one customer; "$v = 1, c = 1$" means that only a vehicle of capacity 1 is employed to deliver the jobs; the last term states the objective being the makespan $C_{\max}$. As the problem is NP-hard, we focus on polynomial time approximation algorithms. We use the worst-case performance ratio to estimate such algorithms. In general, for an instance I of a minimization NP-hard problem, let $C^H(I)$ and $C^*(I)$ denote, respectively, the objective value generated by an approximation algorithm H and the optimal value. The worst-case ratio of H is defined by the largest ratio $C^H(I)/C^*(I)$ for any I. If an algorithm has a worst-case ratio of ρ, then we call it the *ρ-approximation*.

When $m = 1$, that is, when jobs are first processed on a single machine, Chang and Lee [1] showed that the problem is NP-hard and presented a 5/3-approximation algorithm. He et al. [4] presented an improved 53/35-approximation algorithm, while Zhong et al. [8] presented an improved $(3/2 + \epsilon)$-approximation algorithm. Lu and Yuan [5] designed a best possible 3/2-approximation algorithm. When $m = 2$, that is, when jobs are first processed on two parallel (identical) machines, Chang and Lee [1] also presented an approximation algorithm with worst-case performance ratio 2; Zhong et al. [8] presented an improved 5/3-approximation algorithm. Later, Su et al. [7] presented an improved 8/5-approximation algorithm.

In this paper, we study the target problem $(Pm \to D, k = 1 \mid v = 1, c = 1 \mid C_{\max})$. We present an approximation algorithm for the general case when $m \geq 3$. The algorithm has a tight worst-case performance ratio of $7/3 - 1/m$. Note that the target problem in the general case when $m \geq 3$ has not been studied formally in the literature, and it cannot be easily solved by applying the similar methods of dealing with the special cases, $m \leq 2$. Thus, this $(7/3 - 1/m)$-approximation algorithm is the first of such results on this type of approximation algorithm.

Since the bin-packing problem is an important component problem inside the target problem during the job delivery stage, to reduce the number of shipments, we will employ the classical algorithm *first-fit decreasing* (FFD) for the bin-packing problem. In the algorithm FFD, the jobs are sorted into a nonincreasing order of the size, and in this order a job is placed into the lowest indexed batch to which it fits, or otherwise a new batch is created for the job. We use b' to denote the minimum number of batches required in the bin-packing instance *associated* with the target problem; and let b^{FFD} denote the number of batches in the bin-packing solution generated by the algorithm FFD. Let b^* denote the number of batches in an optimal schedule to the target problem. The algorithm FFD has a tight absolute worst-case performance guarantee of $b^{\mathrm{FFD}} \leq 3b'/2$ [6] and a tight asymptotic worst-case performance guarantee of

$b^{\mathrm{FFD}} \leq 11b'/9 + 6/9$ [2]. Clearly, $b' \leq b^*$. We conclude that

$$b^{\mathrm{FFD}} \leq \min\{\tfrac{3}{2}b^*, \tfrac{11}{9}b^* + \tfrac{6}{9}\}. \tag{1.1}$$

2. Approximation algorithm and proof of the worst-case performance ratio

We show in this section that the target problem ($Pm \to D, k = 1 \mid v = 1, c = 1 \mid C_{\max}$), in the general case of $m \geq 3$, admits a $(7/3 - 1/m)$-approximation algorithm, which may be described as follows.

Algorithm 1 H_1

Step 1. Use FFD to pack the jobs of $\mathcal{J}$ by their sizes into batches.
- 1.1. Let k_1 be the total number of resulting batches, denoted as $B_1, B_2, \ldots, B_{k_1}$.
- 1.2. Calculate the sum of processing times for jobs in B_i and denote it by P_i for $i = 1, 2, \ldots, k_1$.
- 1.3. Re-index the batches such that $P_1 \leq P_2 \leq \cdots \leq P_{k_1}$.

Step 2. Sort the jobs within each batch into a nonincreasing order of the processing time.
- 2.1. Starting with B_1, assign each job in B_i to a machine with the least amount of work so far (list scheduling rule [3]) for $i = 1, 2, \ldots, k_1$.

Step 3. Deliver a finished batch as early as possible. If multiple batches have finished when the vehicle becomes available, deliver the batch with the smallest index.

We denote this approximation algorithm as H_1. The achieved schedule by the algorithm is denoted as $\pi_1 = \langle B_1, B_2, \ldots, B_{k_1}\rangle$, which is a sequence of job batches. Note that only after all of the jobs of a batch are finished can the batch be said to be *finished* and ready for transportation to its customer. The makespan of the schedule π_1 is denoted as $C_{\max}^{\pi_1}$. Let ρ_i denote the latest completion time on the machine of the jobs assigned to B_i. Thus, by H_1,

$$C_{\max}^{\pi_1} = \max_{1 \leq i \leq k_1} \{\rho_i + (k_1 - i + 1)T\}. \tag{2.1}$$

Next we estimate the makespan of an optimal schedule for the target problem, denoted as $C_{\max}^*$. Let k^* denote the number of batches in the optimal schedule. For simplicity, the sum of the processing times of all jobs in $\mathcal{J}$ is denoted as P, and let $p_{\max}$ denote the largest processing time of the jobs among $\mathcal{J}$. Without loss of generality, we assume that jobs are processed consecutively on each machine. Secondly, since we have only one transporter, and the processing of all jobs on the m parallel machines needs at least $\max\{P/m, p_{\max}\}$ times, we have the lower bound

$$C_{\max}^* \geq \max\left\{\frac{P}{m} + T, k^*T, p_{\max} + T\right\}. \tag{2.2}$$

https://doi.org/10.1017/S1446181117000190 Published online by Cambridge University Press

LEMMA 2.1. *For the schedule* π_1,

$$\rho_i \le \frac{iP}{mk_1} + \frac{m-1}{m} \min\{p_{\max}, P_i\}$$

for every $i = 1, 2, \ldots, k_1$.

PROOF. Assume that the last job in batch B_i is J_l. Since batches are sequenced in nondecreasing order of P_j for all j, we get $\sum_{j=1}^{i} P_j \le iP/k_1$. Since the list scheduling rule assigns the current job to the first available machine at the earliest possible time, and the jobs in each batch are assigned to machines by this rule in the algorithm,

$$\rho_i \le \frac{\sum_{j=1}^{i} P_j - p_l}{m} + p_l \le \frac{iP}{mk_1} + \frac{m-1}{m} p_l \le \frac{iP}{mk_1} + \frac{m-1}{m} \min\{p_{\max}, P_i\},$$

where the last inequality holds due to $p_l \le \min\{p_{\max}, P_i\}$. This completes the proof. □

Consequently, using Lemma 2.1 and equations (2.1)–(2.2), we conclude that

$$\begin{aligned}
\frac{C_{\max}^{\pi_1}}{C_{\max}^*} &= \frac{\max_{1\le i\le k_1}\{\rho_i + (k_1 - i + 1)T\}}{C_{\max}^*} \\
&\le \frac{\max_{1\le i\le k_1}\{(m-1)p_{\max}/m + iP/mk_1 + (k_1 - i + 1)T\}}{C_{\max}^*} \\
&\le \frac{m-1}{m} + \frac{\max_{1\le i\le k_1}\{iP/mk_1 + iT/k_1 + (k_1 - i + 1 - (m-1)/m - i/k_1)T\}}{C_{\max}^*} \\
&\le \frac{m-1}{m} + \max_{1\le i\le k_1}\left\{\frac{i}{k_1} + \frac{k_1 - i + 1 - (m-1)/m - i/k_1}{k^*}\right\}.
\end{aligned}$$

Therefore,

$$\frac{C_{\max}^{\pi_1}}{C_{\max}^*} \le \frac{m-1}{m} + \max_{1\le i\le k_1}\left\{\frac{i}{k_1} + \frac{k_1 + 1/m - i - i/k_1}{k^*}\right\}. \tag{2.3}$$

LEMMA 2.2. *If* $k_1 \le k^*$, *then* $C_{\max}^{\pi_1}/C_{\max}^* \le 7/3 - 1/m$.

PROOF. If $k_1 < k^*$, then

$$\max_{1\le i\le k_1}\left\{\frac{i}{k_1} + \frac{k_1 + 1/m - i - i/k_1}{k^*}\right\} = \frac{k_1^2 + (k^* - k_1 - 1)k_1 + k_1/m}{k_1 k^*} < 1,$$

since $k^* - k_1 - 1 \ge 0$ and $m \ge 3$. Consequently, equation (2.3) reduces to $C_{\max}^{\pi_1}/C_{\max}^* \le 2 - 1/m$.

If $k_1 = k^*$, then

$$\max_{1\le i\le k_1}\left\{\frac{i}{k_1} + \frac{k_1 + 1/m - i - i/k_1}{k^*}\right\} = \max_{1\le i\le k_1}\left\{\frac{k_1 - i/k_1 + 1/m}{k_1}\right\} \le 1 - \frac{1}{k_1^2} + \frac{1}{k_1} \le \frac{4}{3},$$

since $m \ge 3$. Consequently, equation (2.3) becomes $C_{\max}^{\pi_1}/C_{\max}^* \le 7/3 - 1/m$.

Thus, we have $C_{\max}^{\pi_1}/C_{\max}^* \le 7/3 - 1/m$, which proves the lemma. □

LEMMA 2.3. *If* $k_1 > k^*$ *and* $k^* \le 5$, *then* $C_{\max}^{\pi_1}/C_{\max}^* \le 7/3 - 1/m$.

https://doi.org/10.1017/S1446181117000190 Published online by Cambridge University Press

Proof. If $k_1 > k^*$, we have $k^* \geq 2$ by equation (1.1). We consider the following two cases.

Case 1. Let $k^* = 2$. From equation (1.1), we have $k_1 = 3$; thus, equation (2.3) becomes

$$\frac{C_{\max}^{\pi_1}}{C_{\max}^*} \leq \frac{m-1}{m} + \max_{1\leq i\leq 3}\left\{\frac{i}{3} + \frac{3 - 4i/3 + 1/m}{2}\right\} \leq \frac{m-1}{m} + \frac{4}{3} = \frac{7}{3} - \frac{1}{m},$$

where the second inequality holds due to $m \geq 3$.

Case 2. Let $3 \leq k^* \leq 5$. From equation (1.1), we have $k_1 = k^* + 1$. When $(k^*, k_1) = (3, 4)$, equation (2.3) becomes

$$\frac{C_{\max}^{\pi_1}}{C_{\max}^*} \leq \frac{m-1}{m} + \max_{1\leq i\leq 4}\left\{\frac{i}{4} + \frac{4 - 5i/4 + 1/m}{3}\right\} = \frac{41}{18} - \frac{1}{m},$$

where the last inequality holds due to $m \geq 3$. Similarly, when $(k^*, k_1) = (4, 5)$ or $(k^*, k_1) = (5, 6)$, it follows that

$$\frac{C_{\max}^{\pi_1}}{C_{\max}^*} \leq \frac{67}{30} - \frac{1}{m} \quad \text{or} \quad \frac{C_{\max}^{\pi_1}}{C_{\max}^*} \leq \frac{11}{5} - \frac{1}{m}.$$

Thus, we always have $C_{\max}^{\pi_1}/C_{\max}^* \leq 7/3 - 1/m$, which completes the proof. □

Lemma 2.4. *If $k_1 > k^*$ and $6 \leq k^* \leq 9$, then $C_{\max}^{\pi_1}/C_{\max}^* \leq 7/3 - 1/m$.*

Proof. If $6 \leq k^* \leq 9$, from equation (1.1), we have $k_1 \leq k^* + 2$. When $k_1 = k^* + 1$, equation (2.3) becomes

$$\frac{C_{\max}^{\pi_1}}{C_{\max}^*} \leq \frac{m-1}{m} + \max_{1\leq i\leq k_1}\left\{\frac{k_1 - i/k_1 + 1/m}{k^*}\right\} = \frac{m-1}{m} + \frac{k_1 - 1/k_1 + 1/m}{k^*} \leq \frac{20}{9} - \frac{1}{m},$$

where the first inequality holds due to $k_1 > k^*$ and the last inequality holds due to $k_1 = k^* + 1$, $k^* \geq 6$ and $m \geq 3$. When $k_1 = k^* + 2$, we consider the following two cases.

Case 1. ($m \geq 4$) Since $k_1 = k^* + 2$, $6 \leq k^* \leq 9$ and $m \geq 4$, equation (2.3) becomes

$$\begin{aligned}
\frac{C_{\max}^{\pi_1}}{C_{\max}^*} &\leq \frac{m-1}{m} + \max_{1\leq i\leq k_1}\left\{\frac{k_1^2 + (k^* - k_1 - 1)i + k_1/m}{k_1 k^*}\right\} \\
&= \frac{m-1}{m} + \frac{(k^*+2)^2 - 3 + (k^*+2)/m}{(k^*+2)k^*} \\
&\leq \frac{m-1}{m} + \frac{(k^*+2)^2}{(k^*+2)k^*} \\
&\leq \frac{7}{3} - \frac{1}{m}.
\end{aligned}$$

Case 2. ($m = 3$) From equation (2.1), we further distinguish the following three subcases according to the value of $C_{\max}^{\pi_1}$.

https://doi.org/10.1017/S1446181117000190 Published online by Cambridge University Press

Case 2.1. $C_{\max}^{\pi_1} = \rho_{k_1} + T$. It follows from Lemma 2.1 and equation (2.2) that

$$\frac{C_{\max}^{\pi_1}}{C_{\max}^*} \le \frac{p/m + (2/3)p_{\max} + T}{C_{\max}^*} \le \frac{5}{3}.$$

Case 2.2. $C_{\max}^{\pi_1} = \rho_{k_1-1} + 2T$. From Lemma 2.1,

$$\rho_{k_1-1} \le \frac{\sum_{j=1}^{k_1-1} P_j}{3} + \frac{2}{3}\min\{p_{\max}, P_{k_1-1}\}.$$

Consequently, using $P_{k_1-1} \le P_{k_1}$ and equation (2.2),

$$\begin{aligned}
\frac{C_{\max}^{\pi_1}}{C_{\max}^*} &\le \frac{\left(\sum_{j=1}^{k_1-1} P_j\right)/3 + (2/3)\min\{p_{\max}, P_{k_1-1}\} + 2T}{C_{\max}^*} \\
&\le \frac{\left(\sum_{j=1}^{k_1-1} P_j + P_{k_1-1} + p_{\max}\right)/3 + 2T}{C_{\max}^*} \\
&\le \frac{\left(\sum_{j=1}^{k_1-1} P_j + P_{k_1} + p_{\max}\right)/3 + 2T}{C_{\max}^*} \\
&= \frac{(P + p_{\max})/3 + 2T}{C_{\max}^*} \\
&\le 2.
\end{aligned}$$

Case 2.3. $C_{\max}^{\pi_1} = \max_{1\le i\le k_1-2}\{\rho_i + (k_1 - i + 1)T\}$. From Lemma 2.1, $\rho_i \le (\sum_{j=1}^{i} P_j)/3 + 2P_i/3$. Also, since $P_1 \le P_2 \le \cdots \le P_{k_1}$,

$$\sum_{j=1}^{i} P_j + 2P_i \le \sum_{j=1}^{i+2} P_j \le (i+2)\frac{P}{k_1}$$

when $i \le k_1 - 2$. Consequently, using equation (2.2), $k_1 = k^* + 2$ and $6 \le k^* \le 9$ yields

$$\begin{aligned}
\frac{C_{\max}^{\pi_1}}{C_{\max}^*} &\le \frac{\max_{1\le i\le k_1-2}\left\{\left(\sum_{j=1}^{i} P_j\right)/3 + 2P_i/3 + (k_1 - i + 1)T\right\}}{C_{\max}^*} \\
&\le \max_{1\le i\le k_1-2}\left\{\frac{(i+2)P/3k_1 + (k_1 - i + 1)T}{C_{\max}^*}\right\} \\
&\le \max_{1\le i\le k_1-2}\left\{\frac{(i+2)P/3k_1 + (i+2)T/k_1}{(P/3) + T} + \frac{(k_1 - i + 1)T - (i+2)T/k_1}{k^*T}\right\} \\
&= \max_{1\le i\le k_1-2}\left\{\frac{i+2}{k_1} + \frac{k_1 - i + 1 - (i+2)/k_1}{k^*}\right\} \\
&= \frac{(k^*+2)^2 + 3k^* - 3}{k^*(k^*+2)} \\
&< 2.
\end{aligned}$$

In summary, we always have $C_{\max}^{\pi_1}/C_{\max}^* \le 7/3 - 1/m$, which proves the lemma. □

https://doi.org/10.1017/S1446181117000190 Published online by Cambridge University Press

TABLE 1. An instance I of the problem $(Pm \to D, k = 1 \mid v = 1, c = 1 \mid C_{\max})$ with $m = 3$ and $T = 1$.

Job index j	1	2	3	4	5	6
Processing time p_j	ϵ	ϵ	ϵ	1	1	1
Size s_j	31/60	29/120	29/120	1/3	1/3	9/40

LEMMA 2.5. *If $k_1 > k^*$ and $k^* \geq 10$, then*

$$\frac{C_{\max}^{\pi_1}}{C_{\max}^*} \leq \frac{209}{90} - \frac{1}{m}.$$

PROOF. Inequality (2.3) changes to

$$\frac{C_{\max}^{\pi_1}}{C_{\max}^*} \leq \frac{m-1}{m} + \max_{1\leq i\leq k_1}\left\{\frac{k_1 - i/k_1 + 1/m}{k^*}\right\} \leq \frac{m-1}{m} + \frac{11k^*/9 + 6/9 + 1/3}{k^*} \leq \frac{209}{90} - \frac{1}{m}.$$

Here the first inequality holds due to $k_1 > k^*$, the second inequality holds due to equation (1.1) and $m \geq 3$ and the last inequality holds due to $k^* \geq 10$. This proves the lemma. □

THEOREM 2.6. *For the problem $(Pm \to D, k = 1 \mid v = 1, c = 1 \mid C_{\max})$, in the general case of $m \geq 3$, the algorithm H_1 is a $(7/3 - 1/m)$-approximation.*

PROOF. We prove the theorem by considering all feasible combinations of k^* and k_1 satisfying equation (1.1). All four of the different combinations of k^* and k_1 have been separately dealt with in Lemmas 2.2–2.5. The maximum ratio between the makespan achieved by the algorithm H_1 and the optimal makespan $C_{\max}^*$ among the four lemmas is $7/3 - 1/m$, which completes the proof. □

In addition, we present an instance I of the problem $(Pm \to D, k = 1 \mid v = 1, c = 1 \mid C_{\max})$ in Table 1 to show that the performance ratio $7/3 - 1/m$ of the algorithm H_1 is tight. In the instance I, there are six jobs, three machines and one vehicle with a capacity of 1. A round trip between the machine and the customer is 1 (that is, $T = 1$).

The six jobs in the instance I can be formed into two batches; $B_1^* = \{J_1, J_2, J_3\}$ and $B_2^* = \{J_4, J_5, J_6\}$. Since the processing times of the batches are $P_1^* = 3\epsilon$ and $P_2^* = 3$, by setting the job batch processing order as $\langle B_1^*, B_2^*\rangle$, we achieve a makespan $2 + 2\epsilon$ when $\epsilon > 0$ is sufficiently small. In fact, observe that this is actually an optimal schedule, that is, $C_{\max}^* = 2 + \epsilon$.

On the other hand, the six jobs in the instance I can be formed into three batches $B_1 = \{J_6\}$, $B_2 = \{J_1, J_4\}$ and $B_3 = \{J_5, J_2, J_3\}$ by algorithm H_1 and $\pi_1 = \langle B_1, B_2, B_3\rangle$. Since $P_1 = 1$, $P_2 = 1 + \epsilon$ and $P_3 = 1 + 2\epsilon$, we have $\rho_1 = 1$, $\rho_2 = 1$ and $\rho_3 = 1 + \epsilon$. Consequently, by equation (2.1), we have $C_{\max}^{\pi_1} = 4$. It follows that the performance ratio of the algorithm H_1 on the instance I is $4/(2 + \epsilon) \to 2$ when $\epsilon \to 0$.

https://doi.org/10.1017/S1446181117000190 Published online by Cambridge University Press

3. Conclusions

In this paper, we studied a parallel machine scheduling problem with the finished job delivery to a single customer, $(Pm \to D, k = 1 \mid v = 1, c = 1 \mid C_{\max})$. We presented an algorithm with a tight bound of $7/3 - 1/m$ for the general case when $m \geq 3$. It would be interesting to see whether the problem admits a better approximation algorithm, for example, by using a better bin-packing algorithm or finding some relationship between the job size and its processing times on the parallel machines. In addition, it is worth extending the results to the problem of m uniform machines, which can be denoted as $(Qm \to D, k = 1 \mid v = 1, c = 1 \mid C_{\max})$, and has not been studied even when $m = 2$.

Acknowledgements

J. M. Dong is supported by the National Natural Science Foundation of China Grant No. 11501512 and the Science Foundation of Zhejiang Sci-Tech University (ZSTU) Grant No. 13062171-Y; J. L. Hu is supported by the National Natural Science Foundation of China Grant No. 11471286; and L. L. Wang is supported by the National Natural Science Foundation of China Grant No. 61503109.

References

[1] Y.-C. Chang and C.-Y. Lee, "Machine scheduling with job delivery coordination", *European J. Oper. Res.* **158** (2004) 470–487; doi:10.1016/S0377-2217(03)00364-3.

[2] G. Dósa, R. Li, X. Han and Zs. Tuza, "Tight absolute bound for first fit decreasing bin-packing: $\mathrm{FFD}(L) \leq (11/9)\mathrm{OPT}(L) + 6/9$", *Theoret. Comput. Sci.* **510** (2013) 13–61; doi:10.1016/j.tcs.2013.09.007.

[3] R. L. Graham, "Bounds for certain multiprocessing anomalies", *Bell Syst. Tech. J.* **45** (1966) 1563–1581; doi:10.1002/j.1538-7305.1966.tb01709.x.

[4] Y. He, W. Y. Zhong and H. Gu, "Improved algorithms for two single machine scheduling problems", *Theoret. Comput. Sci.* **363** (2006) 257–265; doi:10.1016/j.tcs.2006.04.014.

[5] L. F. Lu and J. J. Yuan, "Single machine scheduling with job delivery to minimize makespan", *Asia-Pac. J. Oper. Res.* **25** (2008) 1–10; doi:10.1142/S0217595908001596.

[6] D. Simchi-Levi, "New worst-case results for the bin packing problem", *Naval Res. Logist.* **41** (1994) 579–585; doi:10.1002/1520-6750(199406)41:4<579::AID-NAV3220410409>3.0.CO;2-G.

[7] C.-S. Su, J. C.-H. Pan and T.-S. Hsu, "A new heuristic algorithm for the machine scheduling problem with job delivery coordination", *Theoret. Comput. Sci.* **410** (2009) 2581–2591; doi:10.1016/j.tcs.2009.02.019.

[8] W. Y. Zhong, G. Dósa and Z. Y. Tan, "On the machine scheduling problem with job delivery coordination", *European J. Oper. Res.* **182** (2007) 1057–1072; doi:10.1016/j.ejor.2006.09.059.