Theoretical Computer Science 410 (2009) 2581–2591

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

A new heuristic algorithm for the machine scheduling problem with job delivery coordination

Chi-Shiang Su [a], Jason Chao-Hsien Pan [b,*], Tsung-Shin Hsu [a]

[a] Department of Industrial Management, National Taiwan University of Science and Technology, 43 Keelung Road, Section 4, Taipei 106, Taiwan

[b] Department of Business Administration, Takming University of Science and Technology, 56 Huanshan Road, Section 1, Taipei 114, Taiwan

ARTICLE INFO

Article history:
Received 22 October 2008
Received in revised form 15 February 2009
Accepted 19 February 2009
Communicated by D.-Z. Du

Keywords:
Scheduling
Supply chain management
Complexity theory
Heuristic

ABSTRACT

In a rapidly changing environment, competition among enterprises has a tendency towards competing between supply chain systems instead of competing between individual companies. Traditional scheduling models which only address the sequence of jobs to be processed at the production stage under some criteria are no longer suitable and should be extended to cope with the distribution stage after production. Emphasizing on the coordination and integration among various members of a supply chain has become one of the vital strategies for the modern manufacturers to gain competitive advantages. This paper studies the NP-hard problem of the two-stage scheduling in which jobs are processed by two parallel machines and delivered to a customer with the objective of minimizing the makespan ($P2 \to D, k = 1|v = 1, c = z|C_{\max}$). The proposed heuristic algorithm is shown to have a worst-case ratio of 63/40, except for two particular cases.

1. Introduction

Traditional scheduling models only addressed the sequence of jobs to be processed at the production stage under some criteria. Nevertheless, it is no longer suitable and the models should be extended with transportation considerations to cope with the distribution stage after production. Lee and Chen [6] investigated machine scheduling models that impose constraints on both transportation capacity and transportation times, and discussed the computational complexity of various scheduling problems by either showing the NP-hardness or proposing polynomial algorithms for these problems. Chang and Lee [1] further studied the problems in which each job requires different physical space for delivery, whereas Li et al. [7] investigated a problem involving job deliveries to multiple customers at different locations. Lee and Chen [6], and Soukhal et al. [12] analyzed the complexity issues of a class of flow shop problems. He et al. [3] studied a class of single machine with two-stage scheduling problems proposed by Chang and Lee [1] and reduced the worst-case ratio from 5/3 to 53/35. Woeginger [13] studied parallel machine environment with equal job arrival times and proposed a heuristic method with worst-case analysis. Other related researches can be found in [10,14,9,2].

This paper focuses mainly on a class of two parallel machines' problem, in which jobs need to be delivered to a single customer by a vehicle after their production stages. The problem was first proposed by Chang and Lee [1] and can be described as follows: There is a set of n independent jobs, $N = \{J_1, J_2, \ldots, J_n\}$, to be processed without preemption at a manufacturing system consisting of two identical machines, M_1 and M_2. Each job J_i, $i = 1, 2, \ldots, n$, must be first processed in the manufacturing system by one of the two identical machines and has a processing time p_i, and the finished jobs are delivered in batches to the customer by a vehicle. Moreover, a job size s_j, which represents the physical space J_j occupies

* Corresponding author. Tel.: +886 2 2658 5801x5171; fax: +886 2 2658 5801x2731.
E-mail address: jpanbox@gmail.com (J.C.-H. Pan).

0304-3975/$ – see front matter © 2009 Published by Elsevier B.V.
doi:10.1016/j.tcs.2009.02.019

Table 1
A list of notation.

Algorithm (procedure)	$H2$	A	B	C	D	LPT	Optimal schedule
Schedule	σ^{H2}	$\bar{\sigma}$	$\tilde{\sigma}$	$\bar{\sigma}'$	$\bar{\bar{\sigma}}$	σ^L	
The number of batches	b^{H2}	$\bar{b}$	b^{H2}	$\bar{b}$	b^{H2}		b^*
The makespan	C^{H2}	$\bar{C}$	$\tilde{C}$	$\bar{C}'$	$\bar{\bar{C}}$		C^*
The time when machines finish processing the last job	$C(M)$	$\overline{C(M)}$	$\tilde{C}(M)$	$\overline{C(M)}'$	$\overline{\overline{C(M)}}$	$C(M)^L$	$C(M)^*$
The departure time of jobs in the first batch	x	$\bar{x}$		$\bar{x}'$	x		u
The total processing time of jobs in the second batch	y	$\bar{y}$		$\bar{y}'$	y		v
The total processing time of jobs in the third batch							w

when being loaded in the vehicle, is associated with each job. There is only one vehicle initially located at the manufacturing system and available with a limited capacity c representing the total physical space that the vehicle provides for one delivery at the manufacturing facility. Chang and Lee [1] also presented a polynomial time algorithm with a worst-case ratio of 2. Zhong et al. [15] presented an improvement algorithm for the problem and reduced the worst-case ratio to 5/3. The purpose of this paper is to propose a new algorithm which leads to a best possible solution with a worst-case ratio of 63/40, except for the two particular cases below where $C^{MH3}/C^* \leq 8/5$. Table 1 lists the notation defined by Zhong et al. [15] and used in this paper.

(1) $b^* = 3$ and $\bar{b}' = 4$, with $\bar{C}' = \bar{x}' + \bar{b}'T$ or $\bar{C}' = \bar{y}' + (\bar{b}' - 1)T$ (if it exists).
(2) $b^* = 2$ and $\bar{b}' = 3$, with $\bar{C}' = \bar{x}' + \bar{b}'T$ or $\bar{C}' = \bar{y}' + (\bar{b}' - 1)T$ (if it exists).

2. Problem: $P2 \to D, k = 1|v = 1, c = z|C_{\max}$

This section considers the two-stage scheduling problem with a single machine and one customer area, or $P2 \to D, k = 1|v = 1, c = z|C_{\max}$. Let P be the total processing time of all the jobs and t be the one-way transportation time between the machine and the customer. Hence, each delivery has the same transportation time of $T = 2t$.

This problem was first solved by Chang and Lee [1]. They proposed an algorithm $H2$ with a worst-case ratio of 2. Zhong et al. [15] stated that there are two points preventing the worst-case ratio of $H2$ from getting better. The first point is that $H2$ ignores the processing times of jobs when assigning jobs to batches and the second is that $H2$ does not take the idleness of the other machine into account when assigning jobs in one batch to a machine as a whole. Therefore, they proposed an algorithm $MH2$ to improve on these two points, and the algorithm includes procedures A and B that are executed depending on the values of b^{H2} and C^{H2}. The worst-case ratio of $MH2$ is 5/3.

Let the makespan of $\tilde{\sigma}$ be $\tilde{C}$ and the time the machines finish processing the last job be $C(\tilde{M})$. On the other hand, let the resulting schedule be σ^{H2} with makespan C^{H2} and the time the machines finish processing the last job be $C(M)$.

Next, consider one point which prevents the worst-case ratio of $MH2$ from being better. Algorithm $MH2$ assigns batches to machines according to the shortest processing time (*SPT*) rule; hence, the loads over the two machines may not be well balanced and lead to a larger makespan. The longest processing time (*LPT*) rule is a method developed for $Pm\|C_{\max}$ problem [8] and has the effect of balancing the load among various machines. Furthermore, since the $P2\|C_{\max}$ problem is a special case of the $Pm\|C_{\max}$ problem, we can apply the *LPT* rule to the problem and then reverse the sequence batches assigned on each machine such that the batches are sorted in the increasing order of their processing times, and the loads over the two machines may be better balanced.

3. A new algorithm $MH3$

This section describes a heuristic algorithm $MH3$ for solving the problem under study.

Algorithm *MH3*

Step 1: If $b^{H2} = 3$ or $b^{H2} = 4$, run procedure C; otherwise, run procedure D.
Step 2: Output C^{MH3}, stop.

Procedure C:

Step 1. Construct an instance of the knapsack problem as follows: for each job J_j, $j = 1, 2, \ldots, n$, construct an item with profit p_j and size s_j, and let the knapsack capacity be z. Run any *FPTAS* for the knapsack problem with $\varepsilon = 1/5$, and denote by N_1 the set of items put into the knapsack.
Step 2. Assign all jobs in N_1 to the same batch and assign other jobs to batches by algorithm *FF* (First Fit). Let the total number of the resulting batches be $\bar{b}'$.
Step 3. Define P_k as the total processing time of the jobs in the kth batch, $k = 1, 2, \ldots, \bar{b}'$, and denote the kth batch as B_k.
Step 4. Re-index these batches so that $P_1 \geq P_2 \geq \cdots \geq P_{\bar{b}'}$. Jobs in each batch can be sequenced in any arbitrary order and let $S = \{B_1, B_2, \ldots, B_{\bar{b}'}\}$.

Step 5. Let S_1 and S_2, be two sets of batches, which are both initially empty. Let $\overline{P_{s_1}}$ and $\overline{P_{s_2}}$ be the periods of time for processing all the batches in S_1 and S_2, respectively.
Step 6. Set $i = 1$ and $k = 0$.
Step 7. Set $i = i + k$. If $i > \overline{b}'$, go to step 10.
Step 8. Put batch B_i into the set with smaller total processing time between S_1 and S_2. Move B_i to be the first batch of the set and eliminate it from S.
Step 9. Set $k = k + 1$ and go to step 7.
Step 10. Denote the sequences in set S_1 and S_2 as $\overline{\sigma_1}'$ and $\overline{\sigma_2}'$, respectively.
Step 11. Assign batches one by one to machine 1 and machine 2 according to $\overline{\sigma_1}'$ and $\overline{\sigma_2}'$, respectively, except for batch B_1.
Step 12. Assign the jobs in batch B_1 one by one to machines 1 and 2 according to the *LPT* rule.
Step 13. Dispatch each completed but undelivered batch (all jobs in the same batch are assigned to the same machine, except batch B_1) whenever the vehicle becomes available. If multiple batches have been completed when the vehicle becomes available, dispatch the batch with the earliest completed.
Step 14. Let the obtained makespan be C^{MH3} and the time machines finish processing the last job be $\overline{C(M)}'$.

Remark 3.1 (*[15]*). The jobs corresponding to the items in N_1 are assigned to the same batch by algorithm *FF* in Step 2 of procedure *A*.

Remark 3.2 (*[15]*). For the knapsack problem, among others, Lawler [5] proposed an FPTAS with a time complexity of $O(n\log(1/\varepsilon) + 1/\varepsilon^4)$, where $(1-\varepsilon)$ is the worst-case ratio. Kellerer and Pferschy [4] also proposed an FPTAS with a time complexity of $O(n\min\{\log n, \log(1/\varepsilon)\} + (1/\varepsilon^2)\min\{n, (1/\varepsilon)\log(1/\varepsilon)\})$.

Procedure *D*:

Step 1: Assign jobs into batches using *FFD* (First Fit Decreasing) rule. Set the total number of the resulting batches to be b.
Step 2: Define P_k as the total processing time of the jobs in the kth batch, $k = 1, 2, \dots, b$, and denote the kth batch as B_k.
Step 3: Re-index these batches such that $P_b \le P_{b-1} \le \cdots \le P_1$. Jobs in each batch can be sequenced in any arbitrary order, and let $S = \{B_1, B_2, \dots, B_b\}$.
Step 4: Let S_1 and S_2, be the sets of batches, which are both initially empty. Let P_{s1} and P_{s2} be the periods of time for processing all the batches in S_1 and S_2, respectively.
Step 5: Set $i = 1$ and $k = 0$.
Step 6: Let $i = i + k$. If $i > b$, go to step 9.
Step 7: Put batch B_i into the set with smaller total processing time between S_1 and S_2. Move B_i to be the first batch of the set and eliminate it from S.
Step 8: Set $k = k + 1$ and go to step 6.
Step 9: Denote the sequences in set S_1 and S_2 as $\overline{\overline{\sigma_1}}$ and $\overline{\overline{\sigma_2}}$, respectively.
Step10: Assign batches one by one to machines 1 and 2 according to $\overline{\overline{\sigma_1}}$, $\overline{\overline{\sigma_2}}$, respectively, except for batch B_1.
Step 11: Assign the jobs in batch B_1 one by one to machines 1 and 2 according to the *LPT* rule.
Step 12: Dispatch each completed but undelivered batch (all jobs in the same batch are assigned to the same machine, except batch B_1) whenever the vehicle becomes available. If multiple batches have been completed when the vehicle becomes available, dispatch the batch with the earliest completed.
Step 13: Let the resulting makespan be C^{MH3} and the time machines finish processing the last job be $\overline{\overline{C(M)}}$.

Both procedures *C* and *D* schedule batches by *LPT* rule. Therefore, it follows that $C(M)^L \ge \overline{\overline{C(M)}}$ and $C(M)^L \ge \overline{C(M)}'$ since the sequences generated are merely different in the composition of batches.

Next, three examples are presented to illustrate the procedure of the proposed heuristic algorithm.

Example 3.1. Consider the instance given in [15]. Let $T = \delta, z = 2, s_1 = s_2 = s_3 = s_4 = 1, p_1 = p_2 = \delta$ and $p_3 = p_4 = 1$.

First run *H*2. We get $b^{H2} = 2, B_1 = \{J_1, J_2\}, B_2 = \{J_3, J_4\}, P_1 = 2\delta$ and $P_2 = 2$. The jobs are processed and delivered as shown in Fig. 1.

Next, run *MH*2. From algorithm *H*2, we obtain $C(M) = 2$ and $C^{H2} = C(M) + T = 2 + \delta$. Since $C^{H2} = C(M) + T$, run procedure *B* and we have $B_s = \{J_4\}$ and $\tilde{C} = 1 + 3\delta$. The jobs are processed and delivered as shown in Fig. 2.

Then, we run algorithm *MH*3 and get $b^{H2} = 2, B_1 = \{J_1, J_2\}, B_2 = \{J_3, J_4\}, P_1 = 2\delta, P_2 = 2$ and $C^{MH3} = 1 + 3\delta$. The jobs are processed and delivered as illustrated in Fig. 2. According to Zhong et al. [15], it follows that $C^* = 1 + 2\delta$.

Consequently, we have $C^{MH3}/C^* = (1+3\delta)/(1+2\delta) < 63/40$, while $C^{MH2}/C^* < 63/40$ and $C^{MH2}/C^* \le 2$.

Example 3.2. Consider the instance given in [15]. Let $T = 2, z = 7, s_1 = s_2 = 3, s_3 = s_4 = s_5 = s_6 = 2, p_1 = p_3 = 1, p_2 = p_4 = p_5 = \delta$ and $p_6 = 2$.

First run *H*2. We get $b^{H2} = 3, B_1 = \{J_1, J_2\}, B_2 = \{J_3, J_4, J_5\}, B_3 = \{J_6\}, P_1 = 1 + \delta, P_2 = 1 + 2\delta, P_3 = 2$ and $C^{H2} = C(M) + T = 7 + \delta$. The jobs are processed and delivered as shown in Fig. 3.

Next, run *MH*2. Since $b^{H2} = 3$ and $C^{H2} \ne C(M) + T$, *MH*2 goes to step 4 and executes procedure *A*. According to Zhong et al. [15], $\overline{b} = 2$ and $\overline{B_1} = \{J_2, J_4, J_5\}, \overline{B_2} = \{J_1, J_3, J_6\}, \overline{P_1} = 3\delta, \overline{P_2} = 4$ and $C^{MH2} = \min\{C^{H2}, \overline{C}\} = \{7 + \delta, 6\} = 6$. The jobs are processed and delivered as depicted in Fig. 4.

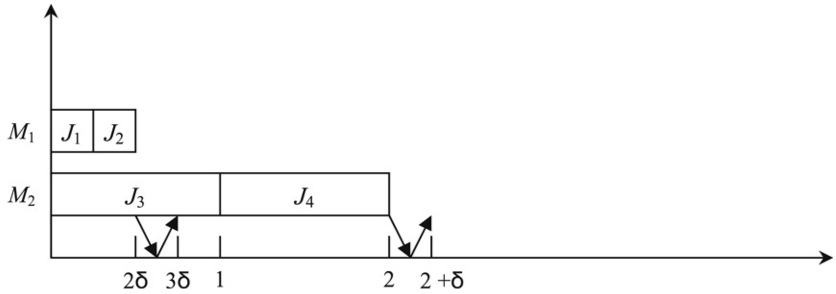

Fig. 1. The results of Example 3.1 scheduled by algorithm *H*2.

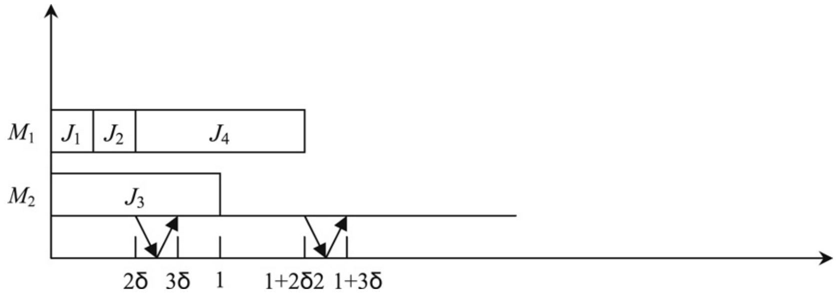

Fig. 2. The results of Example 3.1 scheduled by algorithm *MH*2 or *MH*3.

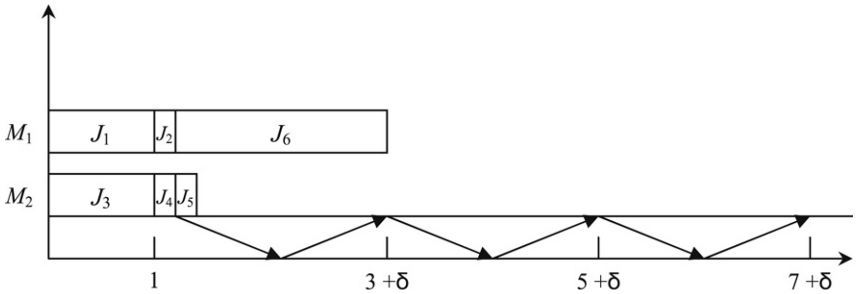

Fig. 3. The results of Example 3.2 scheduled by algorithm *H*2.

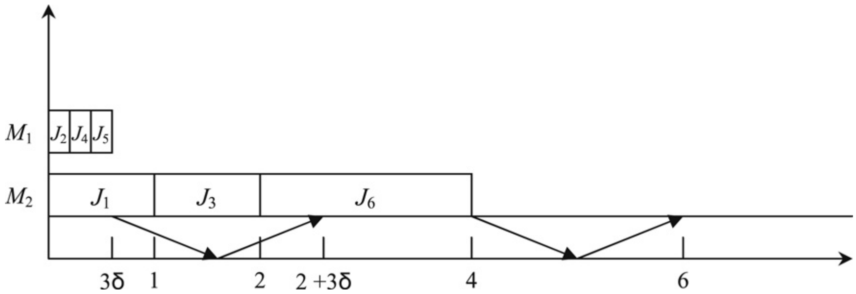

Fig. 4. The results of Example 3.2 scheduled by algorithm *MH*2.

Then, run algorithm *MH*3. Since $b^{H2} = 3$, *MH*3 goes to step 1 and executes procedure *C*. It follows that $\overline{b}' = 2$, $\overline{B_1}' = \{J_1, J_2\}$, $\overline{B_2}' = \{J_3, J_4\}$, $\overline{P_1}' = 2\delta$, $\overline{P_2}' = 2$ and $C^{MH3} = 4 + 3\delta$. The jobs are processed and delivered as illustrated in Fig. 5. According to Zhong et al. [15], $C^* = 4 + 2\delta$.

Consequently, we have $C^{MH3}/C^* = (4 + 3\delta)/(4 + 2\delta) \approx 1$, while $C^{MH2}/C^* \leq 3/2$ and $C^{H2}/C^* \leq 7/4$.

Example 3.3. Consider the instance given in [15]. Let $T = 6$, $z = 7$, $s_1 = s_2 = s_3 = s_4 = 3$, $s_5 = s_6 = s_7 = s_8 = 2$, $s_9 = 1$, $p_1 = p_4 = p_8 = 6$, $p_2 = p_3 = \delta$, $p_5 = p_6 = p_7 = 2$ and $p_9 = 2\delta$.

First run *H*2. We have $b^{H2} = 4$, $B_1 = \{J_8\}$, $B_2 = \{J_5, J_6, J_7\}$, $B_3 = \{J_3, J_4\}$, $B_4 = \{J_1, J_2, J_9\}$, $P_1 = 6$, $P_2 = 6$, $P_3 = 6+\delta$, $P_4 = 6 + 3\delta$ and $C^{H2} = \bar{x} + kT = 6 + 4 \times 24 = 30$. The sequences of jobs processed and delivered are shown in Fig. 6.

Next, run *MH*2. Since $b^{H2} \neq 3$ and $C^{H2} \neq C(M) + T$, *MH*2 goes to step 2 and output C^{H2}. Namely, $C^{MH2} = C^{H2}$.

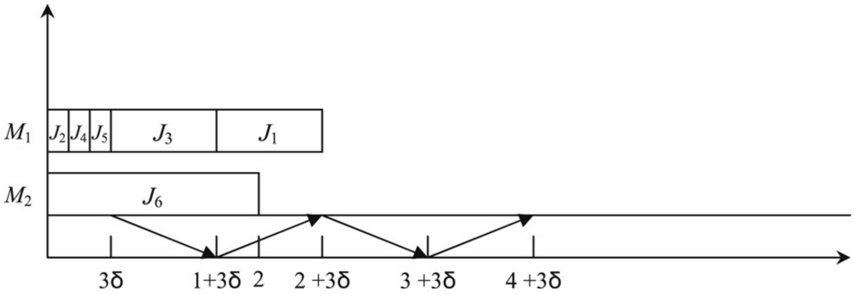

Fig. 5. The results of Example 3.2 scheduled by algorithm *MH*3.

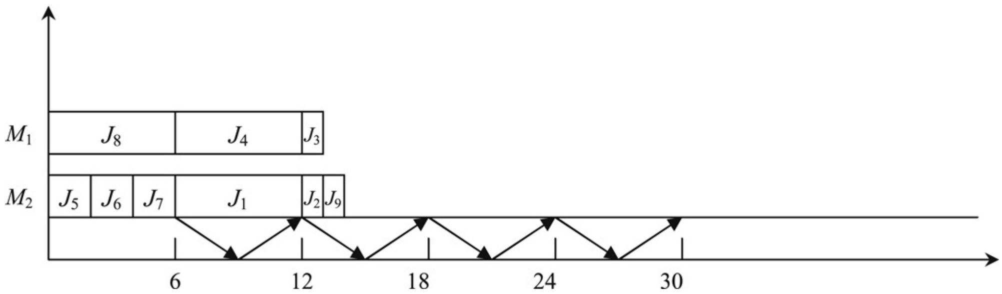

Fig. 6. The results of Example 3.3 scheduled by algorithm *H*2 or *MH*2.

Then, we run algorithm *MH*3. Since $b^{H2} = 4$, *MH*3 goes to step 1 and runs procedure *C*. It follows that $\overline{b}' = 4$, $\overline{B_1}' = \{J_8, J_1, J_5\}$, $\overline{B_2}' = \{J_4, J_6, J_7\}$, $\overline{B_3}' = \{J_9, J_2, J_3\}$, $\overline{P_1}' = 14$, $\overline{P_2}' = 10$, $\overline{P_3}' = 4\delta$ and $C^{MH3} = 22 + 4\delta$. The jobs are processed and delivered as shown in Fig. 7.

According to Zhong et al. [15], we have $C^* = 18 + 2\delta$.

Consequently, it follows that $C^{MH3}/C^* = (22 + 4\delta)/(18 + 2\delta) \leq 63/40$, while $C^{MH2}/C^* \leq 5/3$ and $C^{H2}/C^* \leq 5/3$.

Next, we will prove that except for certain cases, the worst-case performance of algorithm *MH*3 is 63/40.

Theorem 1. *For MH3, we have* $\overline{\overline{C(M)}}/C(M)^* \leq 107/80$*, and* $\overline{C(M)}'/C(M)^* \leq 107/80$.

Proof. Consider the batches scheduled by *LPT* rule, and regard a batch as a job for simplicity. Hence, the problem can be treated as a $P2\|C_{\max}$ [8] and proved by discussing the following conditions:

(1) $P_1 \leq \sum_{i=2}^{b^{H2}} P_i$ and the shortest batch is the last one to finish its processing.

By contradiction, assume that there exists one or more counterexamples where the ratios are strictly larger than 107/80. If more than one such counterexample exists, there must be a problem which has the smallest number of jobs, say n jobs. Since $C(M)^* \geq (\sum_{i=1}^{n} P_i)/2$, we have

$$C(M)^L \leq \frac{\sum_{i=1}^{n-1} P_i}{2} + P_n = \frac{1}{2}P_n + \frac{\sum_{i=1}^{n} P_i}{2} \leq \frac{1}{2}P_n + C(M)^*.$$

The following series of inequalities holds for the counterexample:

$$\frac{107}{80} < \frac{C(M)^L}{C(M)^*} \leq \frac{\frac{1}{2}P_n + C(M)^*}{C(M)^*} \leq 1 + \frac{P_n}{2C(M)^*}$$
$$\frac{27}{80} < \frac{P_n}{2C(M)^*} \Longrightarrow \frac{27}{40}C(M)^* < P_n \Longrightarrow \frac{27}{80}(P_1 + P_2 + \cdots + P_n) < P_n.$$

According to Algorithm *MH*3, we have $P_1 \geq P_2 \geq \cdots \geq P_n$. Therefore, for the smallest counterexample, it implies that the scheduling can result in at most two jobs, namely, P_1 and P_n, i.e., $P_1 \geq P_n > 27/80P$ and $P_1 < 53/80P$.

Thus, we have $\overline{\overline{C(M)}}/C(M)^* \leq C(M)^L/C(M)^* \leq P_1/((1/2P)) = (53/80)P/((1/2)P) < 107/80$.

(2) $P_1 \leq \sum_{i=2}^{b^{H2}} P_i$ and the shortest batch is not the last one to finish its processing.

Delete the shortest batches until a shortest batch is the last one to finish its processing. Consequently, $C(M)^L$ remains the same while $C(M)^*$ may remain the same or decrease. From part (1), we have $\overline{\overline{C(M)}}/C(M)^* \leq C(M)^L/C(M)^* \leq P_1/((1/2P)) = (53/80)P/((1/2)P) < 107/80$.

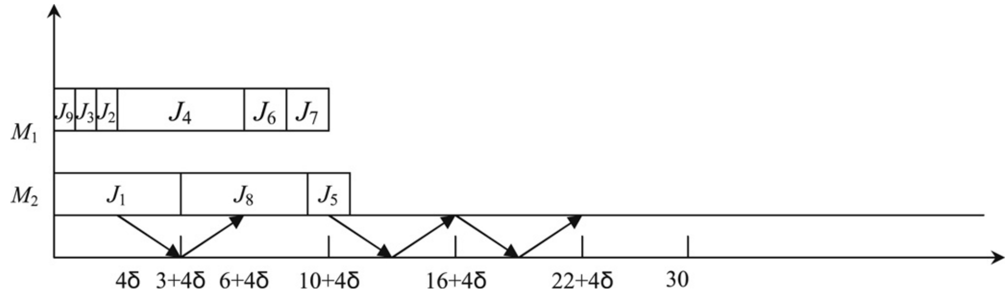

Fig. 7. The results of Example 3.3 scheduled by algorithm *MH*3.

(3) $P_1 > \sum_{i=2}^{b^{H2}} P_i$.

There are two cases to be considered and they are as follows:

Case 1. If batch 1 has only one job, then it is obvious that the schedule is optimal.

Case 2. If batch 1 has more than one job, then from parts (1) and step 11 of procedure *D*, we have $\overline{\overline{C(M)}} \le (\sum_{i=2}^{n} P_i + \sum_{j=2}^{n_1} p_{1_j})/2 + p_{11}$ where $p_{11} = \min\{p_{12}, p_{13}, \ldots, p_{1n1}\}$. It follows that $\overline{\overline{C(M)}}/C(M)^* \le ((\sum_{i=2}^{n} P_i + \sum_{j=2}^{n_1} p_{1_j})/2 + p_{11})/((\sum_{i=2}^{n} P_i + \sum_{j=2}^{n_1} p_{1_j})/2 + p_{11}/2) \le 4/3 < 107/80$.

Therefore, we obtain $\overline{\overline{C(M)}}/C(M)^* \le 107/80$. In a similar way, we can obtain $\overline{C(M)}'/C(M)^* \le 107/80$. □

Lemma 3.1 ([15]). *For an instance I of the bin-packing problem, let OPT(I), FF (I), FFD(I) be the number of used bins in an optimal solution, the solutions yielded by FF and FFD, respectively. We have*

(1) *([11]). FF(I)* $\le (7/4)OPT(I)$.
(2) *([14]). FFD(I)* $\le (11/9)OPT(I) + 1$.

Lemma 3.2. *Denote d as the time period while there is only one machine processing jobs. Thus* $d = \overline{\overline{C(M)}} - (P - \overline{\overline{C(M)}})$, *and let the batch completed last be* B_t. *For any batch k, if* $P_{Bt} \le P_k$, *then* $d \le P_k$, $k = 1, 2, \ldots$.

Proof. Consider the following two conditions:

(1) The shortest batch is the last one to finish its processing.
Recall that $P_1 \ge P_2 \ge \cdots \ge P_n$. Since the shortest batch is the last one to finish, it follows that $d \le P_n \le P_k$, $k = 1, 2, \ldots$.
(2) The shortest batch is not the last one to finish its processing. From part (1), we have $d \le P_{Bt}$. Moreover, since $P_{Bt} \le P_k$, hence, $d \le P_k$, $k = 1, 2, \ldots$. □

Lemma 3.3. *Let* σ_1^r *and* σ_2^r *denote the inverse sequences of* $\overline{\overline{\sigma_1}}$ *and* $\overline{\overline{\sigma_2}}$ *on machines* 1 *and* 2, *respectively. For any two batches h and k processed on different machines, let* μ_h^r *and* μ_k^r *denote the times machines start processing batch h and k under* σ_1^r *and* σ_2^r, *respectively. If* $\rho_k + d \le \rho_h$, *then* $P_k \le P_h$.

Proof. By contradiction, assume that there exist one or more counterexamples where P_k is larger than P_h and $\rho_k + d \le \rho_h$, as shown in Fig. 8. There are two cases to be considered and they are as follows:

(1) Batch B_k is processed on the machine which completes all its processing first. In this case, we have $\mu_k^r = \overline{\overline{C(M)}} - \rho_k - d$ and $\mu_h^r = \overline{\overline{C(M)}} - \rho_h$. Since $\rho_k + d \le \rho_h$, we have $\mu_k^r \ge \mu_h^r$.
(2) Batch B_k is not processed on the machine which completes all its processing first.
In this case, we have $\mu_k^r = \overline{\overline{C(M)}} - \rho_k$ and $\mu_h^r = \overline{\overline{C(M)}} - \rho_h - d$. Since $\rho_k + d \le \rho_h$, we have $\mu_k^r \ge \mu_h^r$.

According to algorithm *MH*3, both σ_1^r and σ_2^r are scheduled by *LPT*; therefore, P_k should be less than or equal to P_h, and batch h starts processing earlier than batch k. This contradiction completes the proof of the lemma. □

Lemma 3.4. *If there exists a batch* B_k *such that* $\delta_k = \rho_k$ *and* $k \ge 3$ *under algorithm MH3, then at least one of the equations* $\delta_{k-i} = \rho_{k-i}$, $\delta_{k-i-1} = \rho_{k-i-1}$, *or* $\delta_{k-i-2} = \rho_{k-i-2}$ *holds, for* $i = 1, 2, \ldots, k-2$.

Proof. Note that $\rho_{k+1} \le \rho_k \le \rho_{k-1}$. The lemma can be proved by considering the various conditions of batches B_{k-1}, B_k and B_{k+1} processed on the machines:

(1) Batches B_{k-1}, B_k and B_{k+1} are all processed on the same machine, while another batch, say B_u, is processed on the other machine.
Since $\delta_k = \rho_k$, it follows that $\delta_{k+1} > \rho_{k+1}$, $\delta_{k+2} > \rho_{k+2}, \ldots, \delta_b = \rho_b$. Consequently, we have $\rho_k - \rho_{k-1} > T$, $P_{k-1} > T$, and hence $\delta_{k-1} = \rho_{k-1}$.

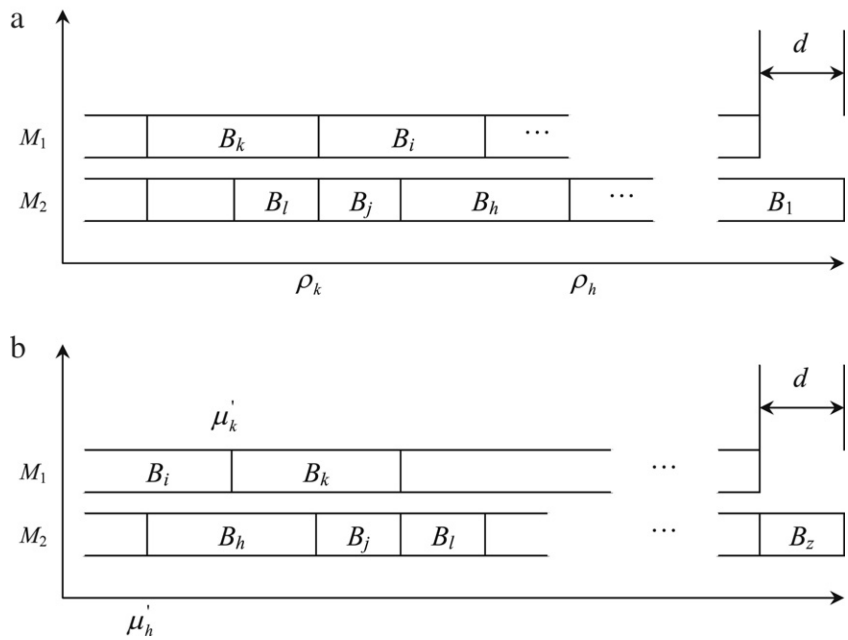

Fig. 8. (a) The schedule by algorithm *MH*3. (b) The schedule by algorithm *LPT*.

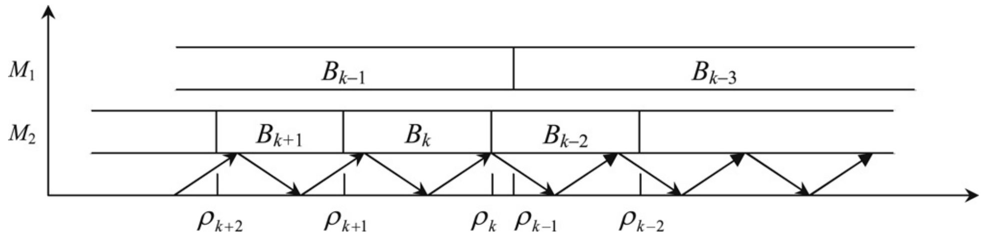

Fig. 9. B_{k3} and B_k are not processed on the same machine.

(2) Both the batches B_k and B_{k-1} are processed on the same machine, while B_{k+1} is processed on the other one.
Since $\delta_k = \rho_k$ and B_{k+1} is processed on the other machine, from Chang and Lee [1], it follows that $P_k > 2T$. Therefore, we have $\delta_{k-1} = \rho_{k-1}$.

When batches B_k and B_{k-1} are not processed on the same machine, it is necessary to consider the various conditions of batches B_{k-2} and B_{k-3} being processed as follows:

(3) Batches B_{k-1} and B_{k-2} are processed on the same machine, while B_k and B_{k+1} are processed on the other one.
If $\rho_{k-1} - \rho_k \geq T$, then $\delta_{k-1} = \rho_{k-1}$.
If $\rho_{k-1} - \rho_k < T$, then from part (2), we know that $P_{k-1} > 2T$ and thus $P_{k-2} > 2T$. Therefore, it follows that $\delta_{k-2} > \rho_{k-2}$.
(4) Batches B_{k+1}, B_k and B_{k-2} are processed on the same machine, while B_{k-1} is processed on the other one.
If $\rho_{k-1} - \rho_k \geq T$, then $\delta_{k-1} = \rho_{k-1}$.
If $\rho_{k-2} - \rho_k \geq 2T$, then $\delta_{k-2} = \rho_{k-2}$.
If $\rho_{k-1} - \rho_k < T$, or $\rho_{k-2} - \rho_k < 2T$, then there two cases to be considered:

Case 1: B_{k-3} and B_k are not processed on the same machine as shown in Fig. 9.
Since $\rho_{k+1} < \rho_k < \rho_{k-1}$, then we have $P_{k-3} \geq P_{k-1} > 3T$ from part (2). Therefore, it follows that $\delta_{k-3} = \rho_{k-3}$.

Case 2: B_{k-3} and B_k are processed on the same machine as shown in Fig. 10.
Since $\rho_k = \delta_k$, we have $P_k \geq P_{k+1} > T$, $P_{k-1} \geq 3T$, and $P_k \geq d$. Meanwhile, from Lemmas 3.2 and 3.3, since batches B_{k-1} and B_{k-3} are processed on different machines and $P_{k-3} \geq P_{k-2} \geq P_k$, we can obtain $\rho_{k-1} + d \leq \rho_{k-1} + P_{k-3} \leq \rho_{k-3}$ and it follows that $3T < P_{k-1} < P_{k-3}$. Thus we have $\delta_{k-3} = \rho_{k-3}$.
(5) B_k and B_{k-2} are processed on the same machine and B_{k-1}, B_{k+1} are processed on the other one.
If $\rho_{k-1} - \rho_k \geq T$, then $\delta_{k-1} = \rho_{k-1}$.
If $\rho_{k-1} - \rho_k < T$, then according to part (2), we know that $P_k > 2T$. Thus $P_{k-2} > 2T$ and it follows that $\delta_{k-2} = \rho_{k-2}$.

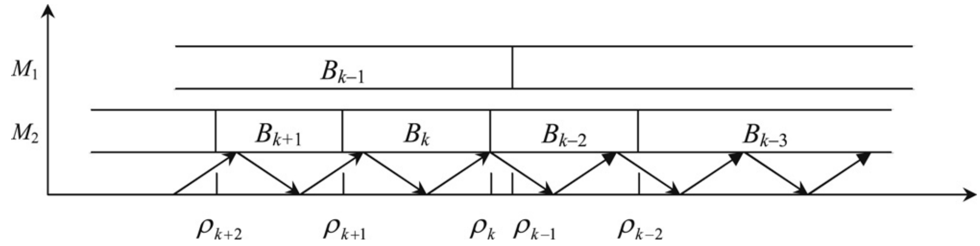

Fig. 10. B_{k3} and B_k are processed on the same machine.

(6) B_{k+1}, B_{k-1} and B_{k-2} are processed on the same machine, and B_k is processed on the other one.
If $\rho_{k-1} - \rho_k \geq T$, then $\delta_{k-1} = \rho_{k-1}$.
If $\rho_{k-1} - \rho_k < T$, then from Lemmas 3.2 and 3.3, since batches B_k and B_{k-2} are processed on different machines, we have $\rho_k + d \leq \rho_k + P_{k-2} \leq \rho_{k-2}$. It follows that $P_k \leq P_{k-2}$. Meanwhile, from part (2), we also know that $2T < P_k$, thus we have $\delta_{k-2} = \rho_{k-2}$.

Therefore, we see that if $\delta_k = \rho_k$, then $\delta_{k-i} = \rho_{k-i}$, or $\delta_{k-i-1} = \rho_{k-i-1}$, or $\delta_{k-i-2} = \rho_{k-i-2}$ holds, for $i = 1, 2, \ldots, k-2$. □

Lemma 3.5. *If $b^{H2} \geq 3$, then $\overline{\overline{C}} \leq \max\{x + b^{H2}T, y + (b^{H2} - 1)T, \overline{\overline{C(M)}} + 2T\}$.*

Proof. If there does not exists a batch B_k on M_1 or M_2 such that $\delta_k = \rho_k$, $k \geq 3$, from Chang et al. [1] and Algorithm *MH*3, we can derive that $\overline{\overline{C}} = \max\{x + b^{H2}T, y + (b^{H2} - 1)T\}$. If there exists a batch B_k such that $\delta_k = \rho_k$, $k \geq 3$, then from Lemma 3.4, we know that $\overline{\overline{C}} \leq \overline{\overline{C(M)}} + 2T$. □

Lemma 3.6 ([1]). $C^* \geq \max\{C(M)^* + T,\ u + b^*T\}$.

Lemma 3.7. *For $b^{H2} \neq 3$ and 4, if $\overline{\overline{C}} = x + b^{H2}T$ or $\overline{\overline{C}} = y + (b^{H2} - 1)T$, then $C^{MH3}/C^* \leq 3/2$.*

Proof. Recall that $b^{H2} \leq (11/9)b_L^* + 1 \leq (11/9)b^* + 1$, and Zhong et al. [15] showed that:

(1)

$$\frac{x + b^{H2}T}{C^*} < \frac{2}{b^{H2}} + \frac{11}{9}\frac{b^{H2} - \frac{2}{b^{H2}}}{b^{H2} - 1} \text{ or } < \frac{2}{b^{H2}} + \frac{b^{H2} - \frac{2}{b^{H2}}}{b^*} \quad (1)$$

(2)

$$\frac{y + b^{H2}T}{C^*} < \frac{2}{b^{H2} - 1} + \frac{11}{9}\frac{b^{H2} - 1 - \frac{2}{b^{H2}-1}}{b^{H2} - 1} \text{ or } < \frac{2}{b^{H2} - 1} + \frac{b^{H2} - 1 - \frac{2}{b^{H2}-1}}{b^*} \quad (2)$$

(3) If $b^{H2} = 1, 2, 5$, or 6, then $C^{H2}/C^* < 63/40$.
Zhong et al. [15] rewrote Eqs. (1) and (2) as

$$f(x) = \frac{2}{x} + \frac{11}{9}\frac{x - \frac{2}{x}}{x - 1} \quad (3)$$

$$g(x) = \frac{2}{x} + \frac{11}{9}\frac{x - \frac{2}{x}}{x}. \quad (4)$$

From Eqs. (3) and (4), we can verify that $f(9) < 63/40$, $f'(x) < 0$ for $x \geq 9$ and $g(8) < 63/40$, $g'(x) < 0$ for $x \geq 8$. Therefore, we only need to show that $C^{H2}/C^* < 63/40$ for $b^{H2} = 7$ or 8.

If $b^{H2} = 7$ and $b^* \geq 6$, then it follows from Eq. (1) that $(x + b^{H2}T)/C^* < 2/7 + (7 - 2/7)/6 < 3/2$, and from Eq. (2) that $(y + (b^{H2} - 1)T)/C^* < 2/6 + (6 - 2/6)/6 < 3/2$.

On the other hand, if $b^{H2} = 8$ and $b^* \geq 7$, it follows from Eq. (1) that $(x + b^{H2}T)/C^* < 2/8 + (8 - 2/8)/7 < 3/2$, and from Eq. (2) that $(y + (b^{H2} - 1)T)/C^* < 2/7 + (7 - 2/7)/7 < 3/2$.

Hence, we have $(x + b^{H2}T)/C^* < 8/5$ and $(y + (b^{H2} - 1)T)/C^* < 8/5$ for $b^{H2} \neq 3, 4$. □

Lemma 3.8. *If $b^* = 2$ and $\overline{b}' = 3$ (if it exists), then $\bar{x}' \leq (1/5)C(M)^* + (4/5)u$.*

Proof. We have $\bar{P}_3 \geq (4/5)P^* \geq (4/5)v$ from [15]. We know that $P \leq 2u + v$, thus $\bar{P}_3 \geq (4/5)(P - 2u)$. Note that $\bar{x}' \leq \bar{P}$, then it follows that

$$\bar{x}' \leq \frac{P - \bar{P}_3}{2} \leq \frac{P - \frac{4}{5}(P - 2u)}{2} = \frac{1}{10}P + \frac{4}{5}u \leq \frac{1}{5}C(M)^* + \frac{4}{5}u. \quad \square$$

Lemma 3.9. *If $b^* = 3$ and $\overline{b}' = 4$ (if it exists), then $\bar{x}' \le (2/15)C(M)^* + (8/15)u + (4/15)v$.*

Proof. It follows from [15] that $\bar{P}_4 \ge (4/5)P^* \ge (4/5)w$. In addition, we have $P \le 2u + v + w$, thus $\bar{P}_4 \ge (4/5)(P - 2u - v)$. Note that $\bar{x}' \le \bar{P}$, hence

$$\bar{x}' \le \frac{P - \bar{P}_4}{3} \le \frac{P - \frac{4}{5}(P - 2u - v)}{3} = \frac{1}{15}P + \frac{8}{15}u + \frac{4}{15}v \le \frac{2}{15}C(M)^* + \frac{8}{15}u + \frac{4}{15}v. \quad \square$$

Lemma 3.10. *If $b^* = 2$, $\overline{b}' \le 3$, $\overline{C(M)}' \ne \bar{x}' + \bar{b}'T$ and $\overline{C(M)}' \ne \bar{y}' + (\bar{b}' - 1)T$ (if it exists), then $C^{MH3}/C^* \le 63/40$.*

Proof. From Theorem 1, we have $\overline{C(M)}'/C(M)^* \le 107/80$.

If $b^* = 2$ and $\overline{b}' = 2$, or $b^* = 2$ together with $\overline{b}' = 3$ and $\overline{C}' = \overline{C(M)}' + T$, then we have

$$\frac{\overline{C}'}{C^*} \le \frac{\overline{C(M)}' + T}{C(M)^* + T} \le \frac{\frac{107}{80}C(M)^* + T}{C(M)^* + T} \le \frac{107}{80} - \frac{\frac{27}{80}T}{C(M)^* + T} < \frac{63}{40}.$$

If $b^* = 2$, $\overline{b}' = 3$ and $\overline{C}' = P - \overline{C(M)}' + 2T$, then we have $T < P - \overline{C(M)}' \le P/2$, and the two cases that need to be considered are as follows:

(1) $C^* = C(M)^* + T \ge u + 2T$

$$\frac{\overline{C}'}{C^*} \le \frac{P - \overline{C(M)}' + 2T}{C(M)^* + T} < \frac{\frac{P}{2} + 2T}{\frac{P}{2} + T} \le 1 + \frac{T}{\frac{P}{2} + T} < \frac{63}{40}.$$

(2) $C^* = u + 2T \ge C(M)^* + T$,

$$\frac{\overline{C}'}{C^*} \le \frac{P - \overline{C(M)}' + 2T}{u + 2T} < \frac{\frac{P}{2} + 2T}{u + 2T} \le \frac{C(M)^* + 2T}{u + 2T} \le \frac{u + 3T}{u + 2T} < \frac{63}{40}. \quad \square$$

Lemma 3.11. *If $b^* = 2$ and $\overline{b}' \le 3$ with $\bar{C}' = \bar{x}' + \bar{b}'T$ or $\bar{C}' = \bar{y}' + (\bar{b}' - 1)T$ (if it exists), then $C^{H2}/C^* \le 8/5$.*

Proof. If $\overline{b}' = 2$ and $b^* = 2$, then from Lemma 3.7, we obtain $\overline{C}'/C^* < 63/40$.

If $\overline{b}' = 3$, since ($\bar{C}' = \bar{x}' + \bar{b}'T$ or $\bar{C}' = \bar{y}' + (\bar{b}' - 1)T$), then from Lemma 3.8, we have $\bar{x}' \le (1/5)C(M)^* + (4/5)u$.

If $u + 2T \ge C(M)^* + T$, then $u + T \ge C(M)^*$ and $\bar{x}' \le (1/5)C(M)^* + (4/5)u \le (1/5)(u + T) + (4/5)u = u + (1/5)T$. Hence,

$$\frac{\overline{C}'}{C^*} \le \frac{\bar{x}' + 3T}{u + 2T} < \frac{u + \frac{1}{5}T + 3T}{u + 2T} = 1 + \frac{\frac{6}{5}T}{u + 2T} \le \frac{8}{5}.$$

If $u + 2T < C(M)^* + T$, then $u + T < C(M)^*$ and $\bar{x}' \le (1/5)C(M)^* + (4/5)u < (1/5)C(M)^* + (4/5)(C(M)^* - T) = C(M)^* - (4/5)T$. Hence,

$$\frac{\overline{C}'}{C^*} \le \frac{\bar{x} + 3T}{C(M)^* + T} < \frac{C(M) * -\frac{4}{5}T + 3T}{C(M)^* + T} = 1 + \frac{\frac{6}{5}T}{C(M)^* + T} \le \frac{8}{5}. \quad \square$$

Lemma 3.12. *If $b^{H2} \ge 4$, $\overline{\overline{C}} \ne x + b^{H2}T$ and $\overline{\overline{C}} \ne y + (b^{H2} - 1)T$, then $C^{MH3}/C^* \le 63/40$.*

Proof. From Theorem 1, we have $\overline{\overline{C(M)}}/C(M)^* \le 107/80$. Recall that *FFD(I)* $\le (11/9)OPT(I) + 1$ [14], so it follows that $b^* \ge 3$.

If $u + b^*T \le C(M)^* + T$ and $b^* \ge 3$, then $C(M)^* \ge (b^* - 1)T$, so

$$\frac{\overline{\overline{C}}}{C^*} \le \frac{\overline{\overline{C(M)}} + 2T}{C(M)^* + T} \le \frac{\frac{107}{80}C(M)^* + 2T}{C(M)^* + T} \le \frac{107}{80} + \frac{\frac{53}{80}T}{C(M)^* + T} \le \frac{107}{80} + \frac{\frac{53}{80}T}{2T + T} < \frac{63}{40}.$$

If $u + b^*T \ge C(M)^* + T$ and $b^* \ge 3$, then $C(M)^* \le u + (b^* - 1)T$, so

$$\frac{\overline{\overline{C}}}{C^*} \le \frac{\overline{\overline{C(M)}} + 2T}{u + b^*T} \le \frac{\frac{107}{80}C(M)^* + 2T}{u + b^*T} \le \frac{107(u + b^*T) + 53T}{80(u + b^*T)} \le \frac{107}{80} + \frac{53T}{80(u + b^*T)} < \frac{63}{40}. \quad \square$$

Lemma 3.13. *According to steps 1 and 2 of procedure C, If $b^* = 3$, then $\bar{b}' \le 4$.*

Proof. Recall that *FF(I)* $\le (7/4)OPT(I)$, thus if $b^* = 3$, then $\bar{b}' \le 5$. Therefore, we only need to show that if $b^* = 3$, then $\bar{b}' \ne 5$.

For simplicity, assume that the bin (batch) capacity, and the size of each job are scaled such that the capacity is 1, and the size of each job is between 0 and 1.

By contradiction, assume that there exists one or more counterexamples where $\bar{b}' = 5$, then we consider the various cases on the size of the first job assigned to batch 5, s_i, as follows:

(1) $s_i \geq 1/2$. According to *FF*, we can conclude that each of the batches B_2, B_3 and B_4 must contain one job whose size is greater than 1/2. Thus, we have $b^* \geq 4$.
(2) $1/2 > s_i \geq 1/3$. It follows that batches B_1, B_2, B_3 and B_4 are all at least $(1 - s_i)$ full. On the other hand, according to step 2 of procedure C, we know that the sizes of all jobs which have been assigned to batches are greater than or equal to s_i. Hence, it follows that $b^* \geq 4$.
(3) $1/3 > s_i > 0$. It implies that $\sum_{i=1}^{b^{H2}} s_i > 3$, thus $b^* \geq 4$.

This contradiction completes the proof of the lemma. □

Lemma 3.14. *If* $b^* = 3$ *and* $\overline{b}' \leq 4$*, with* $\bar{C}' = \bar{x}' + \bar{b}'T$ *or* $\bar{C}' = \bar{y}' + (\bar{b}' - 1)T$ *(if it exists), then* $C^{MH3}/C^* \leq 8/5$.

Proof. If $b^* = 3$ and $\overline{b}' = 4$, with ($\bar{C}' = \bar{x}' + \bar{b}'T$ or $\bar{C}' = \bar{y}' + (\bar{b}' - 1)T$), then from Lemma 3.9, we have $\bar{x}' \leq (2/15)C(M)^* + (8/15)u + (4/15)v$. Furthermore, we have $C^* = \max\{C(M)^* + T, u + 3T\}$ from Lemma 3.6. Therefore, if $u + 3T \geq C(M)^* + T$, then $\bar{x}' \leq (2/15)C(M)^* + (8/15)u + (4/15)v \leq (2/15)(u + 2T) + (8/15)u + (4/15)v = (2/3)u + (4/15)T + (4/15)v$. Hence,

$$\frac{\overline{C}'}{C^*} \leq \frac{\bar{x}' + 4T}{u + 3T} < \frac{\frac{2}{3}u + \frac{64}{15}T + \frac{4}{15}v}{u + 3T} = \frac{2}{3} + \frac{\frac{34}{15}T + \frac{4}{15}v}{u + 3T}.$$

Denote a as the time period of only one machine which is processing the first batch u, then, $a = |d_u^1 - d_u^2|$, where d_u^i denotes completed time of the first batch u on machine i. Therefore, we have $a \leq u$. Since $u + 3T \geq C(M)^* + T$, $\rho_2^* < \delta_2^*$, thus $u + (v - a)/2 \leq \rho_2^* < \delta_2^* = u + T$, and consequently, $v < 2T + a$. Hence, it follows that

$$\frac{\overline{C}'}{C^*} \leq \frac{2}{3} + \frac{\frac{34}{15}T + \frac{4}{15}v}{u + 3T} \leq \frac{2}{3} + \frac{\frac{34}{15}T + \frac{8T+4a}{15}}{u + 3T} \leq \frac{14}{15} + \frac{30T}{15u + 45T} \leq \frac{8}{5}.$$

If $u + 3T < C(M)^* + T$, then $2T < C(M)^* \leq \overline{C(M)}'$ and

$$\frac{\overline{C}'}{C^*} \leq \frac{\bar{x}' + 4T}{C(M)^* + T} \leq \frac{\frac{1}{2}\overline{C(M)}' + 4T}{\frac{27}{20}\overline{C(M)}' + T} = \frac{10}{27} + \frac{1960T}{729\overline{C(M)}' + 540T} < \frac{63}{40}.$$

In a similar way, if $b^* = 3$ and $\overline{b}' = 3$, then we have $\overline{C}'/C^* \leq 63/40$. □

Lemma 3.15. *If* $b^* = 3$, $\overline{b}' \leq 4$, $\overline{\overline{C}} \neq x + b^{H2}T$ *and* $\overline{\overline{C}} \neq y + (b^{H2} - 1)T$ *(if it exists), then* $C^{MH3}/C^* \leq 63/40$.

Proof. It is similar to Lemma 3.12 that if ($b^* = 3$ and $\overline{b}' = 3$) or ($b^* = 3$ and $\overline{b}' = 4$), then $C^{MH3}/C^* \leq 63/40$. □

Theorem 2. $C^{MH3}/C^* \leq 63/40$*, except for the two particular cases below where* $C^{MH3}/C^* \leq 8/5$.
(1) $b^* = 3$ *and* $\overline{b}' = 4$*, with* $\bar{C}' = \bar{x}' + \bar{b}'T$ *or* $\bar{C}' = \bar{y}' + (\bar{b}' - 1)T$ *(if it exists).*
(2) $b^* = 2$ *and* $\overline{b}' = 3$*, with* $\bar{C}' = \bar{x}' + \bar{b}'T$ *or* $\bar{C}' = \bar{y}' + (\bar{b}' - 1)T$ *(if it exists).*

Proof. This is a direct conclusion from Lemmas 3.7, 3.10–3.12, 3.14 and 3.15. □

4. Conclusions and suggestions for future research

This paper studies a two-stage scheduling problem with two parallel machines and one customer to minimize the makespan of jobs ($P2 \to D, k = 1|v = 1, c = z|C_{\max}$). Since the problem is NP-hard, several heuristic methods have been developed and the best worst-case ratio was 5/3. This paper presents a new algorithm based on the *LPT* rule and shows that the proposed method can achieve a worst-case ratio close to 63/40, except for the two particular cases below where the associated ratio is 8/5:
(1) $b^* = 3$ and $\overline{b}' = 4$, with $\bar{C}' = \bar{x}' + \bar{b}'T$ or $\bar{C}' = \bar{y}' + (\bar{b}' - 1)T$ (if it exists).
(2) $b^* = 2$ and $\overline{b}' = 3$, with $\bar{C}' = \bar{x}' + \bar{b}'T$ or $\bar{C}' = \bar{y}' + (\bar{b}' - 1)T$ (if it exists).

Many topics remain for future exploration. First of all, it is worthwhile investigating the complexity of the more general problem $P2 \to D, k = 1|v = 1, c = z|\Sigma C_j$. Secondly, for the intractable $P2 \to D, k = 1|v = k, c = z|\Sigma C_j$ problem, it is justified through developing possibly efficient heuristic algorithms for obtaining approximate solutions. Thirdly, more realistic scheduling models that impose constraints on limited buffer capacities between manufacturing machines, or models involving multiple customer areas with vehicle routing decisions need to be studied. Furthermore, it is interesting to investigate the problems under other performance measures, such as due date related criteria, or under more complicated manufacturing configurations, such as job shop or open shop.

References

[1] Y.C. Chang, Y.L. Lee, Machine scheduling with job delivery coordination, European Journal of Operational Research 158 (2004) 470–487.
[2] T.C.E. Cheng, V.S. Gordon, M.Y. Kovalyov, Single machine scheduling with batch deliveries, European Journal of Operational Research 94 (1996) 277–283.
[3] Y. He, W. Zhong, H. Gu, Improved algorithms for two single machine scheduling problems, Theoretical Computer Science 363 (2006) 257–265.
[4] H. Kellerer, U. Pferschy, A new fully polynomial approximation scheme for the knapsack problem, Lecture Notes in Computer Science 1444 (1998) 123–134.
[5] E. Lawler, Fast approximation algorithms for knapsack problems, Mathematics of Operations Research 4 (1979) 339–356.
[6] C.Y. Lee, Z.L. Chen, Machine scheduling with transportation considerations, Journal of Scheduling 4 (2001) 3–24.
[7] C.L. Li, G. Vairaktarakis, C.Y. Lee, Machine scheduling with deliveries to multiple customer locations, European Journal of Operational Research 164 (2005) 39–51.
[8] M. Pinedo, Scheduling: Theory, Algorithms, and Systems, Prentice-Hall, Englewoods Cliffs, NJ, 1995.
[9] C.N. Potts, M.Y. Kovalyov, Scheduling with batching: A review, European Journal of Operational Research 120 (2000) 228–249.
[10] C.N. Potts, Wassenhove L.N. Van, Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity, Journal of the Operational Research Society 43 (1992) 395–406.
[11] D. Simchi-Levi, New worst-case results for the bin packing problem, Naval Research Logistics 41 (1994) 579–585.
[12] A. Soukhal, A. Oulamara, P. Martineau, Complexity of flow shop scheduling problems with transportation constraints, European Journal of Operational Research 161 (2005) 32–41.
[13] G.J. Woeginger, Heuristics for parallel machine scheduling with delivery times, Acta Informatica 31 (1994) 503–512.
[14] M. Yue, A simple proof of the inequality $FFD(L) \leq 11OPT(L)/9 + 1\forall L$, for the FFD bin-packing algorithm, Acta Mathematica Applicate Sinica 7 (1991) 321–331.
[15] W. Zhong, G. Do'sa, Z. Tan, On the machine scheduling problem with job delivery coordination, European Journal of Operational Research 182 (2007) 1057–1072.