Hybrid Flowshop Scheduling with Interstage Job Transportation

Wei-ya Zhong · Long-hua Lv

Received: 5 January 2014 / Revised: 20 February 2014 / Accepted: 25 February 2014 /

Published online: 27 March 2014

© Operations Research Society of China, Periodicals Agency of Shanghai University, and Springer-Verlag Berlin Heidelberg 2014

Abstract There are a variety of joint job production and transportation scheduling problems that arise in modern manufacturing systems. In this paper, we study one of such problems that arises in a flowshop environment where there are two processing stages and a single transporter that is available to deliver the finished jobs from the first stage to the second. There is a single machine in the first stage and two parallel machines in the second stage. The transporter can carry only one job in each shipment. Each job is first processed on the single machine at stage one, then transported to and processed on one of the two parallel machines at stage two. The objective is to minimize the makespan, i.e., the completion time of the last job in the second stage. Since this problem is strongly NP-hard, we propose a fast heuristic and show that the heuristic has a worst-case bound of $5/2$. We also conduct numerical experiments to evaluate the average performance of this heuristic.

Keywords Hybrid flowshop scheduling · Transportation · Approximation heuristic

Mathematics Subject Classification 90C27 · 90B35 · 68M20

The research was supported by the National Natural Science Foundation of China Grants (No. 11301327).

W. Zhong (✉)

Department of Business Administration, School of Management, Shanghai University, 200444
Shanghai, People's Republic of China
e-mail: wyzhong@shu.edu.cn

L. Lv

Department of Mathematics, Shanghai University, 200444 Shanghai, People's Republic of China

1 Introduction

Production and transportation are two key operational functions in supply chain management. Fierce competition in today's global market and heightened expectations of customers have forced manufacturers to jointly consider production and transportation operations. In the past two decades, integrated production and transportation problems have attracted a rapidly growing interest. According to the transportation function, these problems may be classified into integrated production and interstage distribution scheduling (IPIDS) problems where the transportation function is in the form of job delivery between two consecutive production stages, and integrated production and outbound distribution scheduling (IPODS) problems where the transportation function is in the form of job delivery from the last production stage to the customer sites. In this paper, we consider an IPIDS problem. Such problems are commonly seen in modern manufacturing systems where job processing involves multiple stages and semi-finished jobs are often transported from one stage to the next by robots or automated guided vehicles. Joint consideration of job processing and job delivery in such an environment is necessary in order to achieve a global optimal solution.

In this paper, we consider a scheduling problem as follows: there are two flowshop stages and a job is first processed in the first stage, and then delivered to the second stage for further processing. A single transporter is available for delivering jobs from the first stage to the second and is initially located at the first stage. There is one single machine in the first stage, and two parallel machines in the second stage so that each job only needs to be processed on one of the machines. This processing environment is referred to in the scheduling literature as a hybrid flowshop. The transporter can only carry one job in a shipment between the two stages. The objective of the problem is to minimize the makespan of the jobs which is defined as the completion time of the last job in the second stage.

In the following, we give a brief review of the existing literature related to our problems. Maggu and Das [12] are the earliest scholars who study an IPIDS problem. They consider a two-machine flowshop problem with a sufficient number of transporters so that whenever a job is completed on the first machine it can be transported, with a job-dependent transportation time, to the second machine immediately. They generalize the well-known Johnson's rule [7] to solve their problem to optimality. Langston [11] analyzes some heuristics for a k -station flowshop makespan problem where each station has the same number of machines that can be used to process jobs, and there is only one transporter with a capacity of one to transport jobs with transportation times dependent on the physical locations of the origin and destination machines. Stern and Vitner [16] consider a two-machine flowshop makespan problem where there is only one transporter with a capacity of one. They assume that the transportation times are job-dependent and that there is no intermediate buffer space at either machine so that a job must be held by the transporter if the machine that is to process this job next is currently processing another job. They formulate the problem as an asymmetric traveling salesman problem and give a polynomial-time heuristic. Various two-machine flowshop makespan problems with interstage transporters are also studied by Lee

and Chen [8]. They clarify computational complexity of some of the problems by either proving their NP-hardness or providing polynomial algorithms. Lee and Strusevich [9] study a two-machine flowshop scheduling problem with a single transporter of unlimited capacity and extend the model to the case of openshop machines. They present best possible approximation algorithms with at most two shipments. Gong and Tang [3] study a two-machine flowshop scheduling problem where a single transporter may carry several jobs simultaneously in a shipment. They consider both the case where each job occupies an identical amount of physical space of the transporter and the case where each job occupies a different amount of physical space of the transporter.

Another line of research that is somewhat related to our problem focuses on hybrid flowshop scheduling (HFS) problems, which involve multiple (usually identical) machines in each production stage, compared to a single machine in each stage in classical flowshop scheduling problems. The HFS problems are, in most cases, NP-hard. For instance, the HFS makespan problem, even with two stages only where one stage has two machines and the other stage has a single machine, is strongly NP-hard (Veltman [17]). Heuristic algorithms have been proposed by Gupta [5], Gupta and Tunc [6], Sriskandarajah and Sethi [15], and Chen [1] for solving the two-stage HFS makespan problems when one stage consists of parallel machines and the other stage consists of a single machine. Lee and Vairaktarakis [10] consider two-stage assembly flowshop scheduling problems with parallel machines in the first stage and a single assembly facility in the second stage. They present heuristic algorithms with good worst-case error bounds and show that the average performance of these algorithms is near optimal. Ruiz and Vázquez-Rodríguez [14] review more than 200 papers dealing with the hybrid flowshop problems.

However, there is only one existing study on a hybrid flowshop scheduling problem with interstage job transportation. Naderi et al. [13] study an m -stage hybrid flowshop problem with $m \geq 2$ where there are anticipatory sequence-dependent setup times and job-dependent transportation times in the form of a multi-transporter system. The objective is to minimize total weighted tardiness. They first formulate the problem as a mixed integer linear programming (MILP) model, then present an effective metaheuristic (EMA), and conduct computational experiments to evaluate the performance of the MILP model and EMA. Our problem is a two-stage hybrid flowshop problem with a single transporter. We propose a heuristic for this problem and show that its worst-case bound is $5/2$.

Since we consider IPIDS problems in this paper, we do not review existing literature on IPODS problems. We refer interested readers to a comprehensive survey on IPODS problems by Chen [2].

The remainder of this paper is organized as follows. In Sect. 2, we describe the problem and introduce some preliminary results. In Sect. 3, we present a fast heuristic for the problem and analyze its worst-case bound. In Sect. 4, we conduct numerical experiments to evaluate the average performance of this heuristic. Finally, we conclude the paper in Sect. 5.

2 Problem Descriptions and Preliminaries

Given a job set $N = \{1, 2, \dots, n\}$ to be processed in a two-stage flowshop scheduling environment, each job has to be first processed in stage one, and then transported by a single transporter to stage two for further processing. There is a single machine in stage one and two parallel machines (M_1 and M_2) in stage two. All jobs are available for processing at time 0. The processing times of job j in stage one and stage two are denoted by a_j and b_j , respectively. We denote that $N_1 = \{j | a_j \leq b_j\}$, $N_2 = \{j | a_j > b_j\}$, $A(N) = \sum_{j=1}^n a_j$ and $B(N) = \sum_{j=1}^n b_j$. For simplicity, denote $A(J) = \sum_{j \in J} a_j$ and $B(J) = \sum_{j \in J} b_j$, where J is a subset of N . Pre-emption is not allowed. The transporter is initially located at stage one and can carry only one job in each batch from stage one to stage two. The transportation time from stage one to stage two is t and that from stage two to stage one is τ . The objective is to minimize the makespan, i.e., the maximum job completion time in stage two.

Property 2.1 *For this problem, there exists an optimal schedule in which jobs are processed in stage one without idle time.*

Throughout this paper, we will apply some known results from the literature for parallel-machine, two-machine flowshop, and three-machine flowshop scheduling problems with the objective of minimizing the makespan.

List Scheduling Rule (LS) ([4]) Given a sequence of jobs and two parallel machines, assign the jobs to the machines by the following procedure: from the first job in the sequence, always assign the jobs to the first available machine.

Johnson's Rule ([7]) For a given job set $N = \{1, 2, \dots, n\}$ and a two-machine flowshop, process jobs in N_1 before jobs in N_2 on both machines, with jobs in N_1 sequenced in non-decreasing orders of a_j and jobs in N_2 sequenced in non-increasing orders of b_j . The resulting schedule is optimal for the classical two-machine flowshop makespan problem.

Makespan for Classical Three-Machine Flowshop Problem In our problem, if the transporter is regarded as a machine, then our problem is somewhat similar to the classical three-machine flowshop makespan problem. For the three-machine flowshop makespan problem, Johnson [7] claims that it is sufficient to consider permutation schedules. Given a permutation schedule, denoted as $\sigma = \{1, 2, \dots, n\}$, Johnson [7] shows that the makespan of σ is given as

$$C_{\max}(\sigma) = \max \left\{ \sum_{j=1}^u x_j + \sum_{j=u}^v y_j + \sum_{j=v}^n z_j : 1 \leq u \leq v \leq n \right\}. \quad (2.1)$$

Hence, we have

$$C_{\max}(\sigma) = \sum_{j=1}^u x_j + \sum_{j=u}^v y_j + \sum_{j=v}^n z_j, \quad \text{for some } u, v \text{ with } 1 \leq u \leq v \leq n, \quad (2.2)$$

where x_j , y_j and z_j are the processing times of job j in stages one, two and three, respectively.

3 An Approximation Heuristic

Since even if there is no transportation time between the two stages, our problem is strongly NP-hard ([17]), in this section, we give a heuristic **H** with worst-case bound of $\frac{5}{2}$. In **H**, the jobs are first sequenced by Johnson's rule, and then assigned to the single machine on stage one by list scheduling (LS). When the vehicle is available, transport the jobs in this sequence to stage two, and always assign the jobs to the first available machine. If some conditions are satisfied, output this schedule. Else, move the last finished job at stage two to the first position of the processing sequence.

Algorithm H

Step 1: Sequence the jobs by Johnson's rule and process the jobs on the single machine of stage one according to list scheduling rule. W.l.o.g., assume that the sequence is $\{1, 2, \dots, n\}$. When the vehicle is available, transport the jobs to stage two after being finished on stage one in sequence of $\{1, 2, \dots, n\}$ and assign the jobs in the same sequence to the first available machine on stage two. Let the obtained schedule be σ_1 with makespan $C(\sigma_1)$. By Eq. (2.1), we assume that

$$C(\sigma_1) = \sum_{i=1}^u a_i + k(t + \tau) - \tau + b_v + B(J_v),$$

where $1 \leq u \leq v \leq n$, $1 \leq k \leq v - u + 1$, and J_v is the set of jobs following job v on the same machine on stage two (See Fig. 1. W.l.o.g., we assume that job v is processed on M_1 at stage two). Let the makespan on M_2 at stage two be $C(M_2) = \sum_{i=1}^{u_{M_2}} a_i + k_{M_2}(t + \tau) - \tau + b_{v_{M_2}} + B(J_{v_{M_2}})$, where $1 \leq u_{M_2} \leq v_{M_2} \leq n$, $1 \leq k_{M_2} \leq v_{M_2} - u_{M_2} + 1$, and $J_{v_{M_2}}$ is defined as the same meaning as J_v .

Step 2: If one of the following conditions is satisfied, output σ_1 and stop. Else, go to Step three.

- (1) $v \in N_2$;
- (2) $v \in N_1$, and $b_v + B(J_v) \leq \frac{3}{4}B(N)$;
- (3) $v \in N_1$, $b_v + B(J_v) > \frac{3}{4}B(N)$, and $b_{v_{M_2}} + B(J_{v_{M_2}}) + b_x \leq \frac{3}{4}B(N)$, where x is the last job on M_1 .

Step 3: Process the jobs on the single machine at stage one according to LS rule in the sequence of $\{x, 1, 2, \dots, x - 1, x + 1, \dots, n\}$. Transport the jobs and process them on stage two in the similar way as in Step one. Let the obtained schedule be σ_2 with makespan $C(\sigma_2)$. Output σ_2 and stop.

Denote the obtained schedule be σ^H with objective value of C^H .

Before analyzing the performance of Algorithm **H**, we list the lower bound of the objective value of an optimal schedule C^* .

$$C^* \geq \max\{A(N) + t, \frac{1}{2}B(N) + t, \min_{j \in N}\{a_j\} + n(t + \tau) - \tau, \max_{j \in N}\{a_j + t + b_j\}\}.$$

Lemma 3.1 *If the output schedule is σ_1 , then $C(\sigma_1) \leq \frac{5}{2}C^*$.*

Proof We prove this lemma according to the cases given in Step two.

(1) $v \in N_2$.

(1.1) If $v > u$, then

$$\begin{aligned} C(\sigma_1) &= \sum_{i=1}^u a_i + k(t + \tau) - \tau + b_v + B(J_v) \\ &\leq \sum_{i=1}^u a_i + k(t + \tau) - \tau + a_v + A(J_v) \\ &\leq A(N) + n(t + \tau) - \tau \\ &\leq 2C^*. \end{aligned}$$

(1.2) If $v = u$, then

$$\begin{aligned} C(\sigma_1) &= \sum_{i=1}^u a_i + t + b_u + B(J_v) \\ &\leq \sum_{i=1}^u a_i + t + a_u + A(J_v) \\ &\leq A(N) + t + a_u \\ &\leq 2C^*. \end{aligned}$$

(2) $v \in N_1$, and $b_v + B(J_v) \leq \frac{3}{4}B(N)$.

(2.1) If $u \neq 1$, note that $a_2 + a_3 + \cdots + a_u \geq (u-1)(t + \tau)$ and $a_u \geq a_{u-1} \geq \cdots \geq a_2$, then $a_u \geq (t + \tau)$ and $u = v$. It is easy to verify that

$$\begin{aligned} C(\sigma_1) &= \sum_{i=1}^u a_i + t + b_u + B(J_u) \\ &\leq A(N) + t + \frac{3}{4}B(N) \\ &= A(N) + t + \frac{3}{2} \left[\frac{1}{2}B(N) \right] \\ &\leq \frac{5}{2}C^*. \end{aligned}$$

(2.2) If $u = 1$, note that $a_1 \leq a_2 \leq \cdots \leq a_v$, then $C^* \geq a_1 + k(t + \tau) - \tau$ and

$$\begin{aligned} C(\sigma_1) &= a_1 + k(t + \tau) - \tau + b_v + B(J_v) \\ &\leq a_1 + k(t + \tau) - \tau + \frac{3}{4}B(N) \leq C^* + \frac{3}{2} \left[\frac{1}{2}B(N) \right] \leq \frac{5}{2}C^*. \end{aligned}$$

(3) $v \in N_1$, $b_v + B(J_v) > \frac{3}{4}B(N)$ and $b_{v_{M_2}} + B(J_{v_{M_2}}) + b_x \leq \frac{3}{4}B(N)$. From Fig. 1, we can see that $C(\sigma_1) \leq C(M_2) + b_x$.

(3.1) $C(M_2) = a_1 + k_{M_2}(t + \tau) - \tau + b_{v_{M_2}} + B(J_{v_{M_2}})$, where $k_{M_2} \geq 2$ and $v_{M_2} \geq 2$.

(3.1.1) If $v_{M_2} \in N_2$, then $a_x > b_x$ and

$$\begin{aligned} C(M_2) + b_x &= a_1 + k_{M_2}(t + \tau) - \tau + b_{v_{M_2}} + B(J_{v_{M_2}}) + b_x \\ &\leq a_1 + k_{M_2}(t + \tau) - \tau + a_{v_{M_2}} + A(J_{v_{M_2}}) + a_x \\ &\leq A(N) + n(t + \tau) - \tau \\ &\leq 2C^*. \end{aligned}$$

(3.1.2) If $v_{M_2} \in N_1$, then $a_1 + k_{M_2}(t + \tau) - \tau \leq C^*$ and

$$\begin{aligned} C(M_2) + b_x &= a_1 + k_{M_2}(t + \tau) - \tau + b_{v_{M_2}} + B(J_{v_{M_2}}) + b_x \\ &\leq a_1 + k_{M_2}(t + \tau) - \tau + \frac{3}{4}B(N) \\ &\leq C^* + \frac{3}{2} \left[\frac{1}{2}B(N) \right] \\ &\leq \frac{5}{2}C^*. \end{aligned}$$

(3.2) $C(M_2) = \sum_{i=1}^{u_{M_2}} a_i + t + b_{u_{M_2}} + B(J_{u_{M_2}})$, we can see that

$$\begin{aligned} C(M_2) + b_x &= \sum_{i=1}^{u_{M_2}} a_i + t + b_{u_{M_2}} + B(J_{u_{M_2}}) + b_x \\ &\leq A(N) + t + \frac{3}{2} \left[\frac{1}{2}B(N) \right] \\ &\leq \frac{5}{2}C^*. \end{aligned}$$

We can conclude that $\frac{C(\sigma_1)}{C^*} \leq \frac{C(M_2) + b_x}{C^*} \leq \frac{5}{2}$.

Lemma 3.2 If the output schedule is σ_2 , then $\frac{C(\sigma_2)}{C^*} \leq 2$.

Proof If the output schedule is σ_2 , then $b_v + B(J_v) > \frac{3}{4}B(N)$ and $b_{v_{M_2}} + B(J_{v_{M_2}}) + b_x > \frac{3}{4}B(N)$. Since $x \in J_v$, $b_x > \frac{1}{2}B(N)$. Note that in σ_2 , the processing sequence on stage one is $\{x, 1, 2, \dots, x-1, x+1, \dots, n\}$.

- (1) $C(\sigma_2) = a_x + t + b_x + B(J_x)$, where J_x is defined in the same way as J_v (See Fig. 2). Since $b_x > \frac{1}{2}B(N)$, we can see that $B(J_x) < \frac{1}{2}B(N) < C^*$ and $C(\sigma_2) \leq C^* + C^* = 2C^*$.
- (2) $C(\sigma_2) = a_x + k(t + \tau) - \tau + b_{v_2} + B(J_{v_2})$, where $2 \leq k \leq v_2 \leq n$, and J_{v_2} is defined in the same way as J_v (See Fig. 3). Note that $b_{v_2} + B(J_{v_2}) < \frac{1}{2}B(N) < b_x$, then $C(\sigma_2) \leq a_x + k(t + \tau) - \tau + b_x \leq 2C^*$.
- (3) $C(\sigma_2) = a_x + a_1 + \cdots + a_{u_2} + k(t + \tau) - \tau + b_{v_2} + B(J_{v_2})$, where $1 \leq u_2 < v_2 \leq n$, $2 \leq k \leq v_2 - u_2 + 1$ and $u_2, v_2 \neq x$ (See Fig. 4).

(3.1) If $v_2 \in N_2$, then

$$\begin{aligned} C(\sigma_2) &= a_x + a_1 + \cdots + a_{u_2} + k(t + \tau) - \tau + b_{v_2} + B(J_{v_2}) \\ &\leq a_x + a_1 + \cdots + a_{u_2} + k(t + \tau) - \tau + a_{v_2} + A(J_{v_2}) \\ &\leq A(N) + n(t + \tau) - \tau \\ &\leq 2C^*. \end{aligned}$$

(3.2) If $v_2 \in N_1$, then by the same analysis as case 2 of Lemma 3.1, we know that $u_2 = 1$. Note that $b_{v_2} + B(J_{v_2}) < \frac{1}{2}B(N) < b_x$, $a_1 = \min_{j \in N_1} \{a_j\}$ and $a_1 + k(t + \tau) - \tau \leq C^*$, we can see that

$$\begin{aligned} C(\sigma_2) &= a_x + a_1 + k(t + \tau) - \tau + b_{v_2} + B(J_{v_2}) \\ &\leq a_x + b_x + a_1 + k(t + \tau) - \tau \\ &\leq 2C^*. \end{aligned}$$

- (4) $C(\sigma_2) = a_x + \sum_{i=1}^{u_2} a_i + t + b_{u_2} + B(J_{u_2})$, where J_{u_2} is defined in the same way as J_v (See Fig. 5). Since $b_{u_2} + B(J_{u_2}) < \frac{1}{2}B(N) < b_x$, then $C(\sigma_2) \leq A(N) + t + \frac{1}{2}B(N) \leq 2C^*$.

Summarizing all the above cases, we can conclude that $\frac{C(\sigma_2)}{C^*} \leq 2$. $\square$

Theorem 3.1 *The worst-case ratio of Algorithm H is $\frac{5}{2}$.*

Proof Combining Lemmas 3.1 and 3.2, the worst-case bound of algorithm H is $\frac{5}{2}$. $\square$

4 Computational Results

From the worst-case ratio analysis for algorithm H, we can conclude that in most cases, the performance of this algorithm is much better than $5/2$. If there is a job j with $b_j > \frac{1}{2}B(N)$, then we can guarantee that the worst-case ratio of algorithm H is less than 2. Meanwhile, we also observe that it is not easy to obtain an instance to show that this bound is tight. In this section, we conduct numerical experiments to evaluate the average performance of algorithm H. Besides, we also present two

Fig. 1 Schedule σ_1 **Fig. 2** Schedule σ_2 , where $C(\sigma_2) = a_x + t + b_x + B(J_x)$ **Fig. 3** Schedule σ_2 , where $C(\sigma_2) = a_x + k(t + \tau) - \tau + b_{v_2} + B(J_{v_2})$ **Fig. 4** Schedule σ_2 , where $C(\sigma_2) = a_x + a_1 + \dots + a_{u_2} + k(t + \tau) - \tau + b_{v_2} + B(J_{v_2})$

heuristics, based on the longest processing time first (LPT) rule (Denoted as **H(LPT)**) and Johnson's rule (Denoted as **H(J)**), respectively, to compare the average performance of these heuristics.

Fig. 5 Schedule σ_2 , where $C(\sigma_2) = a_x + a_1 + \dots + a_{u_2} + t + b_{v_2} + B(J_{v_2})$

Algorithm H(LPT): Sequence the jobs s.t. $b_1 \geq b_2 \dots \geq b_n$, and process the jobs on the single machine of stage one according to LS rule. When the vehicle is available, transport the jobs to stage two after being finished on stage one in sequence of $\{1, 2, \dots, n\}$ and assign the jobs in the same sequence to the first available machine on stage two.

Denote the obtained schedule be $\sigma^{H(LPT)}$ with objective value of $C^{H(LPT)}$.

Algorithm H(J): Sequence the jobs by Johnson's rule and process the jobs on the single machine of stage one according to List Scheduling rule. W.l.o.g., assume that the sequence is $\{1, 2, \dots, n\}$. When the vehicle is available, transport the jobs to stage two after being finished on stage one in sequence of $\{1, 2, \dots, n\}$ and assign the jobs in the same sequence to the first available machine on stage two.

Denote the obtained schedule be $\sigma^{H(J)}$ with objective value of $C^{H(J)}$.

The numerical experiments are carried out by Matlab R2011b. In order to reflect the advantage of Algorithm H, $\max\{b_i\}$ is set to be much larger than the other jobs' stage two processing times. For $n = 5, 10, 15, 20, 25, 30, 35, 40, 45, 50$, the processing times a_i and b_i , and the transportation times t and τ are randomly generated in $[1, 10]$ obeying uniform distribution. For each combination of n and $\max\{b_i\} \in \{100, 200, 300\}$, 20 instances are generated. For these 20 instances, when each heuristic of $\{\mathbf{H(LPT)}, \mathbf{H(J)}, \mathbf{H}\}$ is applied, we define

MAX be the maximal value of $\frac{\text{the objective value of the heuristic's solution}}{\text{the optimal objective value}}$,

MIN be the minimal value of $\frac{\text{the objective value of the heuristic's solution}}{\text{the optimal objective value}}$,

AVG be the average value of $\frac{\text{the objective value of the heuristic's solution}}{\text{the optimal objective value}}$.

The results are list in Table 1. We can find that in most cases, Algorithm H is better than H(LPT) and H(J). However, these values are far less than $5/2$, therefore, we have the reason to believe that the worst-case ratio of Algorithm H is much better than $5/2$. This indicates a research direction worth trying.

Table 1 Numerical experiments

Job's processing time	n	Transportation time	H(LPT)			H(J)			H		
			MAX	MIN	AVG	MAX	MIN	AVG	MAX	MIN	AVG
$a_i \in [1, 10]$ $i \in \{1, 2, \dots, n\}$	$n = 5$	$t \in [1, 10]$ $\tau \in [1, 10]$	1	1	1	1.357 1	1	1.112 4	1	1	1
	$n = 10$		1.061 5	1	1.019 2	1.483 9	1	1.217 5	1.061 5	1	1.020 4
	$n = 15$		1.057 9	1	1.022 1	1.463 1	1	1.166 1	1.266 7	1	1.034 5
	$n = 20$		1.037 7	1	1.013 4	1.306 3	1	1.038 5	1.165 3	1	1.023 1
	$n = 25$		1.071 4	1	1.023 1	1.539 7	1	1.037 7	1.539 7	1	1.037 7
	$n = 30$		1.037 7	1	1.014 3	1.179 9	1	1.015 1	1.179 9	1	1.015 1
	$n = 35$		1.061 9	1	1.017	1.057 5	1	1.005 7	1.057 5	1	1.005 7
	$n = 40$		1.039 3	1	1.010 4	1	1	1	1	1	1
	$n = 45$		1.042 4	1	1.011 6	1.008 5	1	1.000 8	1.008 5	1	1.000 8
	$n = 50$		1.065 6	1	1.011 2	1.016 4	1	1.000 8	1.016 4	1	1.000 8
$b_i \in [1, 10]$ $i \in \{1, 2, \dots, n-1\}$ $b_n = 100$	$n = 5$	$t \in [1, 10]$ $\tau \in [1, 10]$	1	1	1	1.264 8	1	1.074 5	1	1	1
	$n = 10$		1	1	1	1.425 2	1	1.128 2	1	1	1
	$n = 15$		1.038 8	1	1.005	1.628 4	1.049	1.265 6	1.038 8	1	1.005
	$n = 20$		1.034 3	1	1.004 4	1.502 3	1	1.254 9	1.034 3	1	1.004 5
	$n = 25$		1.023 7	1	1.010 4	1.282 8	1	1.107 2	1.023 7	1	1.011 1
	$n = 30$		1.052 6	1	1.013 8	1.356 8	1	1.098 5	1.026 7	1	1.012
	$n = 35$		1.033 1	1	1.013 5	1.388 4	1	1.080 8	1.315 2	1	1.032 1
	$n = 40$		1.048 5	1.002	1.014 9	1.242 3	1	1.017	1.242 3	1	1.014 6
	$n = 45$		1.036 5	1	1.012 5	1.292	1	1.026 1	1.292	1	1.026 1
	$n = 50$		1.032 3	1	1.010 6	1.399 2	1	1.036 5	1.399 2	1	1.036 5
$a_i \in [1, 10]$	$n = 5$		1	1	1	1.085 7	1	1.038 5	1	1	1
	$n = 10$		1	1	1	1.278 8	1	1.109	1	1	1
	$n = 15$		1	1	1	1.436 3	1.045	1.158 9	1	1	1

Table 1 continued

Job's processing time	Transportation time	n	H(LPT)			H(J)			H		
			MAX	MIN	AVG	MAX	MIN	AVG	MAX	MIN	AVG
$i \in \{1, 2, \dots, n\}$	$\tau \in [1, 10]$	$n = 20$	1.015 9	1	1.001 2	1.640 1	1.061	1.229 1	1.019 1	1	1.001 4
		$n = 25$	1.018 5	1	1.003 9	1.650 9	1.046	1.262 4	1.018 5	1	1.003 9
$b_i \in [1, 10]$	$\tau \in [1, 10]$	$n = 30$	1.016 9	1	1.004 8	1.461 8	1	1.232 9	1.016 9	1	1.004 8
		$n = 35$	1.023 7	1	1.010 8	1.467 9	1	1.195	1.023 7	1	1.010 5
$i \in \{1, 2, \dots, n-1\}$	$\tau \in [1, 10]$	$n = 40$	1.020 6	1	1.010 2	1.389 6	1	1.146 2	1.020 6	1	1.010 1
		$n = 45$	1.014	1	1.005 7	1.299	1	1.072 5	1.014	1	1.005 7
$b_n = 300$		$n = 50$	1.018	1	1.008 2	1.463	1	1.094	1.018	1	1.007 3

5 Conclusions

In this paper, we have studied a two-stage hybrid flowshop scheduling problem with interstage job transportation. There is one single machine in the first stage and two parallel machines in the second stage. A single transporter initially located at the first stage can carry one job in a shipment between the two stages. Each job is first processed on the single machine at stage one, then transported to and processed on one of the parallel machines at stage two. The objective is to minimize the maximum completion time of the jobs. We propose a fast heuristic with worst-case bound of $5/2$ and conduct numerical experiments to evaluate the average performance of this heuristic.

Regarding future research, one can search for a heuristic with better worst-case bound and can consider more complex situations where there are multiple parallel machines on both stages and multiple transporters that can transport jobs between the two stages. Obviously, it will be a challenge to find fast heuristics for such problems that have a constant worst-case bound.

References

- [1] Chen, B.: Analysis of classes of heuristics for scheduling two-stage flow shop with parallel machines at one stage. *J. Oper. Res. Soc.* **46**, 234–244 (1995)
- [2] Chen, Z.-L.: Integrated production and outbound distribution scheduling: review and extensions. *Oper. Res.* **58**, 130–148 (2010)
- [3] Gong, H., Tang, L.: Two-machine flowshop scheduling with intermediate transportation under job physical space consideration. *Comput. Oper. Res.* **38**, 1267–1274 (2011)
- [4] Graham, R.L.: Bounds for certain multiprocessing anomalies. *Bell Syst. Tech. J.* **45**, 1563–1581 (1966)
- [5] Gupta, J.N.D.: Two-stage, hybrid flowshop scheduling problem. *J. Oper. Res. Soc.* **39**, 359–364 (1988)
- [6] Gupta, J.N.D., Tunc, E.A.: Schedules for a two-stage hybrid flowshop with parallel machines at the second stage. *Int. J. Prod. Res.* **29**, 1489–1502 (1991)
- [7] Johnson, S.M.: Optimal two- and three-machine production schedules with setup times included. *Nav. Res. Log.* **1**, 61–68 (1954)
- [8] Lee, C.Y., Chen, Z.-L.: Machine scheduling with transportation considerations. *J. Sched.* **4**, 3–24 (2001)
- [9] Lee, C.Y., Strusevich, V.A.: Two-machine shop scheduling with an uncapacited interstage transporter. *IIE Trans.* **37**, 725–736 (2005)
- [10] Lee, C.Y., Vairaktarakis, G.L.: Performance comparison of some classes of flexible flow shops and job shops. *Int. J. Flex. Manuf. Syst.* **10**, 379–405 (1998)
- [11] Langston, M.-A.: Interstage transportation planning in the deterministic flow-shop environment. *Oper. Res.* **35**, 556–564 (1987)
- [12] Maggu, P.L., Das, G.: On $2 \times n$ sequencing problem with transportation times of jobs. *Pure Appl. Math. Sci.* **12**, 1–6 (1980)
- [13] Naderi, B., Zandieh, M., Shirazi, M.A.H.A.: Modeling and scheduling a case of flexible flowshops: total weighted tardiness minimization. *Compu. Ind. Eng.* **57**, 1258–1267 (2009)
- [14] Ruiz, R., Vázquez-Rodríguez, J.A.: The hybrid flow shop scheduling problem. *Eur. J. Oper. Res.* **205**, 1–18 (2010)
- [15] Sriskandarajah, C., Sethi, S.P.: Scheduling algorithms for flexible flowshops: worst and average case performance. *Eur. J. Oper. Res.* **43**, 143–160 (1989)
- [16] Stern, H.I., Vitner, G.: Scheduling parts in a combined production-transportation work cell. *J. Oper. Res. Soc.* **41**, 625–632 (1990)
- [17] Veltman, B.: Multiprocessor scheduling with communication delays. PhD Thesis, Centrum Wiskunde & Informatica, Amsterdam, The Netherlands (1993)