Int J Adv Manuf Technol (2013) 65:787–802
DOI 10.1007/s00170-012-4217-x

ORIGINAL ARTICLE

A mathematical model and extension algorithm for assembly flexible flow shop scheduling problem

Parviz Fattahi · Seyed Mohammad Hassan Hosseini · Fariborz Jolai

Received: 28 August 2011 / Accepted: 1 May 2012 / Published online: 19 May 2012

Abstract In this paper, a scheduling problem for a two-stage production system including machining operations and assembly operations is studied. In this system, a number of products of different kinds are produced. Each product is assembled with a set of several parts. The first stage is a hybrid flow shop to produce parts. All machines can process all kinds of parts in this stage, but each machine can process only one part at a time. The second stage is a single assembly machine or a single assembly team of workers. The considered objective is to minimize the completion time of all products (makespan). A mathematical modeling is presented, and since this problem has been proved strongly nondeterministic polynomial-time hard, a series of heuristic algorithms based on the basic idea of Johnson algorithm is proposed. Also, two lower bounds is introduced and improved to evaluate the final solution obtained from heuristic algorithms. The numerical experiments are used to evaluate the performance of the proposed algorithms.

Keywords Scheduling · Hybrid flow shop · Assembly · Heuristic · Two-stage production system

P. Fattahi (✉)
Department of Industrial Engineering, Bu-Ali Sina University, Hamedan, Iran
e-mail: fattahi@basu.ac.ir

S. M. H. Hosseini
Department of Industrial Engineering, Payame Noor University, Tehran, Iran
e-mail: h.hosseini@phd.pnu.ac.ir

F. Jolai
Department of Industrial Engineering, University of Tehran, Tehran, Iran
e-mail: fjolai@ut.ac.ir

1 Introduction

The two-stage assembly scheduling problem has many applications in industry and hence has received increasing attention from researchers recently [1, 2]. Lee et al. [1] described an application in a fire engine assembly plant while Potts et al. [3] described an application in personal computer manufacturing. In particular, manufacturing of almost all items may be modeled as a two-stage assembly scheduling problem [2]. Hence, after Lee et al. published their paper about three-machine assembly-type flow shop scheduling problem in 1993, the assembly scheduling problems have received considerable attention from researchers during the last two decades.

In these production systems, usually, there is a fabrication stage and an assembly stage. The fabrication stage that can be a single machine, flow shop, hybrid flow shop, or job shop processes and fabricates the parts (components) independently. The assembly stage performs assembly (joining) operations of the product. The main criterion for this problem is the minimization of the maximum job completion time (makespan) [4].

In this paper, a two-stage assembly scheduling problem is studied in which the first stage (fabrication) is a hybrid flow shop. The hybrid flow shop (HFS) is also called flexible flow shop, compound flow shop, and multiprocessor flow shop. It is a generalization of the flow shop in such a way that every job can be processed by one among several machines on each machine stage [5–7]. Hybrid flow shops are common manufacturing environments in which a set of n jobs are to be processed in a series of m stages optimizing a given objective function. There are a number of variants of hybrid flow shop, but all of them have most of the following characteristics in common [8]:

1. The number of processing stages (m) is at least 2.
2. Each stage k has $M^{(k)}$ machines in parallel format ($M^{(k)} \geq 1$ for all stages and $M^{(k)} > 1$ for at least on stage).

3. All jobs are processed following the same production flow: stage 1, stage 2,…, stage m. A job might skip any number of stages provided it is processed in at least one of them.
4. Each job j requires a processing time P_{jk} in stage k.

In the standard form of the HFS problem, all jobs and machines are available at time zero, machines at a given stage are identical, any machine can process only one operation at a time, and any job can be processed by only one machine at a time; setup times are negligible, preemption is not allowed, the capacity of buffers between stages is unlimited, and problem data are deterministic and known in advance. This machine environment is often referred to as a flexible flow shop, compound flow shop, multiprocessor flow shop, or hybrid flow shop [7, 8].

The HFS problem is, in most cases, nondeterministic polynomial-time (NP)-hard. For instance, HFS is restricted to two processing stages, even in the case when one stage contains two machines and the other one a single machine is NP-hard, after the results of Gupta [8]. Then, it is obvious that our problem (with a more complex structure) is NP-hard.

The first study in assembly-type flow shop scheduling problem was done by Lee et al. in 1993. They studied a two-stage assembly flow shop scheduling problem with makespan objective function. They considered a simple problem in which each product is assembled from two types of parts. The first component of each product must be processed on the first machine, and the second component is processed on the second machine. Finally, the third machine assembles the two parts into a product. They proved that the problem is strongly NP-complete and identified several special cases of the problem that can be solved in polynomial time and suggested a branch and bound solution and also three heuristics [1]. Potts et al. [3] extend the problem to the case of multiple fabrication machines in which there are m machines and one machine at the first and second stages, respectively. They develop a heuristic algorithm with a worst-case ratio bound to minimize the makespan. Hariri and Potts [9] also studied the same problem as [3] and proposed a branch and bound algorithm. Cheng and Wang [10] consider minimizing the makespan in the two-machine flow shop scheduling with a special structure, develop several properties of an optimal solution, and obtain optimal schedules for some special cases. In their model, the first machine produces two types of parts, unique components and common components. The unique components are processed individually, the common components are processed in batches, and a setup is needed to form each batch. The second machine assembles components into products.

Yokoyama [11] studied a hybrid scheduling for the production system including parts machining and assembly operations. In his study, several products of different kinds are ordered to be produced, parts for the products are manufactured in a flow shop, and each product is produced by hierarchical assembly operations from the parts. The parts are assembled into the first subassembly, and several other parts and the first subassembly are assembled into the second subassembly. These assembly operations are continued until the last subassembly that is the final product is obtained. In his model, the objective is minimum weighted sum of completion time of each, and decision variables are the sequence of products to be assembled and the sequence of parts to be processed. He introduces a branch and bound with two lower bounds that can solve problems until 10 products and 15 parts.

Sun et al. [12] studied three-machine, assembly-type flow shop scheduling. They tell that this problem has been proved strongly NP-complete and so propose a series of heuristic algorithms based on the basic idea of Johnson's algorithm and Gupta's idea. The heuristic algorithms can solve all of the worst cases which cannot be solved by the existing heuristic.

Yokoyama and Santos [13] considered flow shop scheduling with assembly operations. In their models, several products of different kinds are ordered, and parts are manufactured. Each part for the products is processed on machine M_1 (the first stage) and then processed on machine M_2 (the second stage). Each product is processed (e.g., joined) with the parts by one assembly operation on assembly stage M_A (the third stage). The objective function is the same as [11]. They developed a solution procedure to obtain an ε_optimal solution based on a branch and bound method.

Allahverdi and Al-Anzi [2] studied a two-stage assembly scheduling problem where there are m machines at the first stage and an assembly machine at the second stage. In their model, the setup times are treated as separate from processing times. They prove that this problem is NP-hard and therefore present a dominance relation and propose three heuristics that are a hybrid tabu search, a self-adaptive differential evolution (SDE), and a new self-adaptive differential evolution (NSDE). They show that the NSDE is the best heuristic for the problem even if setup times are ignored.

Al-Anzi and Allahverdi [14] also considered the same problem as [2] where setup times are ignored. They prove that this problem is NP-hard and therefore propose heuristics based on tabu search (Tabu), particle swarm optimization (PSO), and SDE along with the earliest due date and Johnson (JNS) heuristics to solve the problem. A computational experiment reveals that both PSO and SDE are much superior to tabu. Moreover, it is statistically shown that PSO performs better than SDE.

The remaining sections of this paper are organized as follows: In Section 2, the problem is described completely, and the mathematical model is presented. The solving approach, heuristic algorithms and two improved lower bounds, is presented in Section 3. Design of problems and experiments is described in Section 4 while the evaluation of these heuristics and analysis of result are done in Section 5. Finally, a summary of the work and direction for future research are given in Section 6.

2 Problem description

This paper considers a two-stage assembly scheduling problem containing a hybrid flow shop stage and an assembly stage. Suppose that several products of different kinds are ordered. Each product needs a set of components (parts) to complete, and the parts are processed on a two-stage hybrid flow shop. The objective is to minimize the completion time for all products (makespan). Decision variables are the sequence of products to be assembled and also the sequence of parts and assigning them to machines in each stage of hybrid flow shop to be processed.

One of the applications of this problem is in body making of the car manufacturing industry. As it is shown in Fig. 1, a car-making manufactory generally contains the units of production engine, chassis, and body. The unit of body making includes press shop, assembly, and painting. The press shop that produces some parts such as doors and roofs has usually a flow shop or hybrid flow shop format.

Figure 2 shows a schematic view of a hybrid production system containing a flexible flow shop followed by an assembly section. The inputs containing raw material, parts, or unfinished products are processed on the hybrid flow shop stage. When the set parts of a product are complete, they are joined at the assembly stage. Typically, buffers are located between stages to store intermediate products [6, and it is supposed that there is no limit in buffer storages. The number of machines at hybrid flow shop stages is free, and it can be no equivalent at two stages.

2.1 Notations

We introduce the following notation for this problem:

H	Total number of products
h	Product index (h=1, 2, …, H)
n	Total number of parts
j	Part index (j=1, 2, …, n)
n_h	Total number of parts of product h (h=1,2, .., H)
J_h	Set of indices of parts for product h
l	Stage index (l=1, 2)
P_{lj}	Processing time of machining operation for part j in stage l (l=1, 2)
K_l	Number of parallel machine in stage l
k	Machine index
A_h	Assembly time of product h
M	A very big and positive amount

Also, variables of the mathematical model are as follows:

x_{ijkl}	1, if job j is processed directly after job i on machine k in stage l; 0 otherwise
x_{0ikl}	1, if job i is the first job on machine k in stage l; 0 otherwise
x_{i0kl}	1, if job i is the last job on machine k in stage l; 0 otherwise
$C_j^{(l)}$	Completion time of job j in stage l
F_h	Finish time of the parts for the hth product and ready to assemble
$S_{h'h}$	0, if all parts of product h' are ready to be assembled before the parts of product h; a positive amount otherwise
C_h	Completion time of assembly of product h

The problem is to decide the sequence of products and their parts, and the objective function of the problem is expressed as:

$$Z = \max(C_h)$$

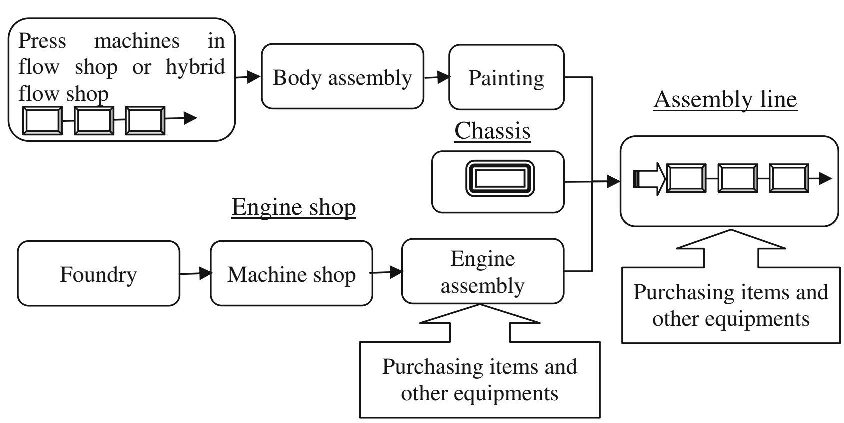

Fig. 1 A car-making manufactory in general

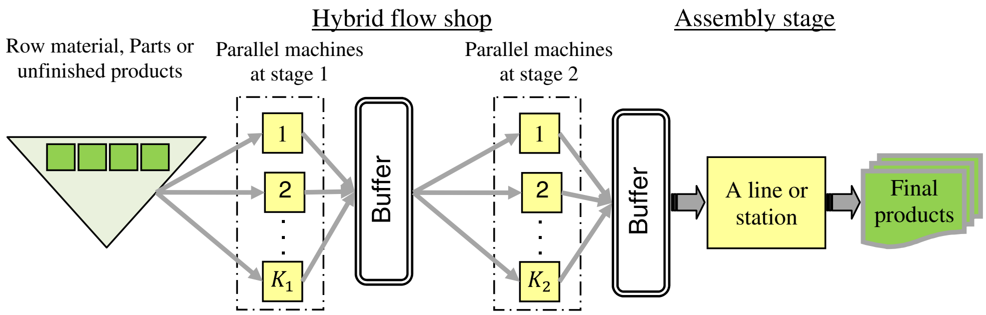

Fig. 2 A schematic view of the considered problem

2.2 Assumptions

1. All parts are available at time zero.
2. The parallel machines in a stage are uniform. That is, the time of machining operation in a stage is the same for all parallel machines about each part.
3. If product h is going to be assembled before product h', then, on each stage, processing of any part for product h' does not start before the start of processing of all parts for product h.
4. Assembly operation for a product will not start until all parts of its product are completed.
5. When assembly operation of a product is started, it does not stop until completed (no preemption in assembly stage).
6. There is no limit in buffer storages.

2.3 Numerical example

In order to clarify the problem, consider a simple numerical example as Table 1. Assume there are two machines in stage 1 and three machines in stage 2. The total number of products is H=4. The data for parts and their processing time of machining operation and assembly are given in Table 1. At first, we scheduled the product and assign their parts to machines according to their number (as a random schedule); the result is shown in Fig. 3 with value Z=62. Then, we solved this problem by the algorithms that are introduced in Section 3 of this paper. The result is shown in Fig. 4 with value Z=52. The numbers in cells represent part indices in Figs. 3 and 4.

The results in Figs. 3 and 4 indicate that the proposed model improves objective function over 16 %. This comparison is done in the condition that assumption 3 is regarded in Fig. 3. That is, if product h is going to be assembled before product h′, then, on each stage, processing of any part for product h′ does not start before the start of the processing of all parts for product h. The solutions show that if this rule was lost on random solution, the difference between results would be more than 16 %. In addition, the idle time of machines in hybrid flow shop has been decreased more than 13 %, and the idle time of the assembly stage has been decreased more than 35 %.

2.4 Mathematical modeling

Based on the present problem and notations, a mathematical formulation for the problem is presented as follows:

$$\text{Min } Z = (C_{\max}) \tag{1}$$

Subject to:

$$\sum_{i=0, i\neq j}^{n} \sum_{k=1}^{K_l} x_{ijkl} = 1 \quad j = 1,2,3,\ldots,n \quad l = 1,2 \tag{2}$$

Table 1 Processing time of machining and assembly

Stage (l)	Products and parts										
	Product 1			Product 2		Product 3			Product 4		
	j=1	j=2	j=3	j=4	j=5	j=6	j=7	j=8	j=9	j=10	j=11
Stage 1	10	8	6	2	4	12	14	10	4	4	4
Stage 2	4	2	4	4	4	8	8	8	4	6	6
Assembly	6			10		8			10		

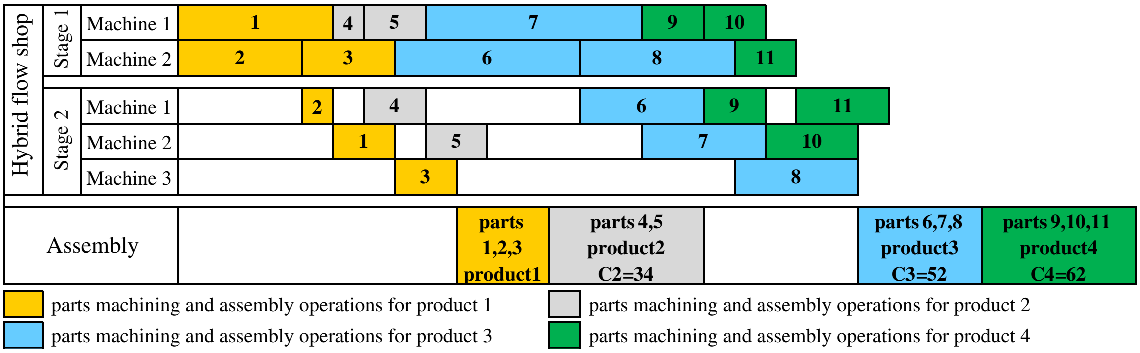

Fig. 3 A random schedule for numerical example with $C_{\max}=62$

$$\sum_{j=0}^{n} x_{ijkl} \leq 1 \quad i = 0, 1, 2, 3, \ldots n \quad l = 1, 2 \quad k = 1, 2, \ldots, K_l \tag{3}$$

$$\sum_{i=0, i\neq h}^{n} x_{ihkl} - \sum_{j=0, j\neq h}^{n} x_{hjkl} = 0 \quad h = 1, 2, 3, \ldots, n \quad l = 1, 2 \quad k = 1, 2, \ldots, K_l \tag{4}$$

$$C_i^{(l)} + \sum_{k=1}^{K_l} x_{ijkl} \times P_{lj} + \left(\sum_{k=1}^{K_l} x_{ijkl} - 1 \right) \times M \leq C_j^{(l)} \quad i = 1, 2, 3, \ldots, \mathrm{n} \quad j = 1, 2, 3, \ldots, \mathrm{n} \quad l = 1, 2 \tag{5}$$

$$C_j^{(1)} + P_{2j} \leq C_j^{(2)} \quad j = 1, 2, 3, \ldots, n \tag{6}$$

$$C_j^{(2)} \leq F_h \quad \forall j \in \{J_h\}\ ,\ h = 2, 3, 4, \ldots, H \tag{7}$$

$$F_h + A_h \leq C_h \quad h = 1, 2, 3, \ldots, H \tag{8}$$

$$C_{h'} + A_h - S_{h'h} \times M \leq C_h \quad h, h' = 1, 2, 3, \ldots, H \tag{9}$$

$$F_{h'} - F_h \leq S_{h'h} \quad h, h' = 1, 2, 3, \ldots, H\ , h' \neq h \tag{10}$$

$$S_{h'h} \geq 0 \quad h, h' = 1, 2, 3, \ldots, H\ , h' \neq h \tag{11}$$

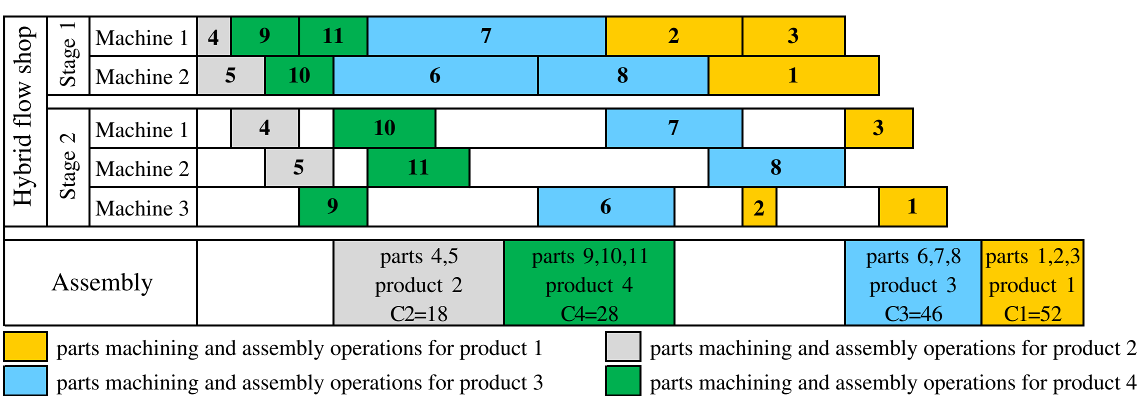

Fig. 4 The best schedule of the numerical example with $C_{\max}=52$

$$C_h \leq C_{\max} \quad h = 1, 2, 3, \ldots, H \tag{12}$$

$$x_{ijkl} \in \{0, 1\} \quad i = 1, 2, 3, \ldots, n\ ;\ j = 1, 2, 3, \ldots, n\ ;\ l = 1, 2 \\ k \leq 1, 2, \ldots, K_1 \tag{13}$$

$$C_j^{(l)} \geq 0 \quad j = 1, 2, 3, \ldots, n \quad l = 1, 2 \tag{14}$$

The makespan minimization aspect of the problem is expressed by Eq. (1). Constraints (2), (3), and (4) ensure that each part is processed precisely once at each stage. In particular, constraint (2) guarantees that at each stage l for each part J_j, there is a unique machine such that either J_j is processed first or after another job on that machine. The inequalities (3) imply that at each stage, there is a machine on which a part has a successor or is processed last. Finally, at each stage for each part, there is one and only one machine satisfying both of the previous two conditions by (4) [5].

Constraints (5) and (6) take care of the completion times of the parts at stages 1 and 2. Inequalities (5) ensure that the completion times C_i^l and C_j^l of parts i and j scheduled consecutively on the same machine respect this order. Inequality (6) implies that parts go through the stages in the right order, i.e., from stage 1 to stage 2. Inequalities (7) take care of the start times of the products at assembly stage. The inequalities (8, 9, 10, and 11) express the completion time of products. Inequalities (9, 10, and 11) ensure that the completion time of product h and h' scheduled consecutively on the assembly stage respects this order.

The constraint that the makespan is not smaller than the completion time of any product is expressed by constraints (12). The last two constraints specify the domains of the decision variables.

3 The proposed solving approach

The decision variables in HFS problem with an assembly stage are the sequence of products to be assembled and the assigning of the parts to machines in each stage, for all products to be processed. Thus, this problem consists of two subproblems. The first subproblem is determining the sequence of products to assemble and the second is assigning the parts of product h to a machine in each stage (for h=1, 2, …, H).

According to assumption 3, if product h is going to be assembled before product h', then, on each stage, processing of any part for product h' does not start before starting the processing of all parts for product h. This assumption is considered because of many reasons such as reducing the inventory costs, damage risk of transportations, and also to facilitate the material flows.

According to these subproblems, we introduce two steps of solving:

Step 1: Find a sequence for the product by the extension of the Johnson algorithm using algorithms A, B, C, D.

Step 2: Determine the sequence and schedule for the parts of any product using methods I, II, III, IV, V, and VI.

3.1 Extension of the Johnson algorithm to determine the sequence of the product

In this section, the Johnson algorithm is extended and used to solve the problem. So, we assume the hybrid flow shop as first stage (or first machine) and assembly as second stage (or second machine). Then, the Johnson extended algorithm is used to determine the sequence of products. So, in this section, only the algorithms of sequencing products are considered.

Table 2 Description of 24 heuristic algorithms

Schedule products	Schedule parts					
	Method I: nondecreasing sum of process time at stages 1 and 2 of hybrid flow shop	Method II: nondecreasing process time at stage 1 of hybrid flow shop	Method III: nondecreasing process time at stage 2 of hybrid flow shop	Method IV: nonincreasing sum of process time at stages 1 and 2 of hybrid flow shop	Method V: nonincreasing process time at stage 1 of hybrid flow shop	Method VI: nonincreasing process time at stage 2 of hybrid flow shop
Algorithm A: Johnson using Eq. 10	AI	AII	AIII	AIV	AV	AVI
Algorithm B: Johnson using Eq. 11	BI	BII	BIII	BIV	BV	BVI
Algorithm C: Johnson using Eq. 12	CI	CII	CIII	CIV	CV	CVI
Algorithm D: Johnson using Eq. 13	DI	DII	DIII	DIV	DV	DVI

Table 3 The batches of problems

Problems types	PT1	PT2	PT3
Interval of index I	$I<1$	$I>1$	$I\approx 1$
Bottleneck	The hybrid flow shop will be bottleneck	The assembly stage will be bottleneck	There will be a balanced condition

The process time of the second stage is assembly time of products and it is shown by A_h (that means assembly time of product h). In the hybrid flow shop stage, the process time of a product will be computed by four equations presented below.

That means that the process time of the first stage consists of the process operation of parts in the hybrid flow shop and it is calculated for any product. This time is shown by PT_h (that means process time of the set parts for product h), and it will be computed by four algorithms, named A, B, C, and D, as shown below:

Calculating methods of PT_h:

$$\text{Algorithm } A : \mathrm{PT}_h = \sum_{l=1}^{2}\sum_{j\in J_h} P_{lj} \quad \text{For } h = 1,2,\ldots,H \tag{15}$$

$$\text{Algorithm } B : \mathrm{PT}_h = \frac{\sum_{l=1}^{2}\sum_{j\in J_h} P_{lj}}{n_h} \quad \text{For } h = 1,2,\ldots,H \tag{16}$$

$$\text{Algorithm } C : \mathrm{PT}_h = \sum_{l=1}^{2}\frac{\sum_{j\in J_h} P_{lj}}{\min(n_h,K_l)} \quad \text{For } h = 1,2,\ldots,H \tag{17}$$

$$\text{Algorithm } D : \mathrm{PT}_h = \max_{j\in J_h}\left\{\max\left(\sum\left(P_{1j}+P_{2j}\right)\right), \left(\frac{\sum P_{1j}}{K_1}+\min\left(P_{2j}\right)\right), \left(\frac{\sum P_{2j}}{K_2}+\min\left(P_{1j}\right)\right)\right\} \quad \text{For } h = 1,2,\ldots,H \tag{18}$$

After computing the PT_h based on the four presented algorithms, the problem will be solved using the Johnson algorithm to determine the sequence of the product. In the second step, we will find the sequence and schedule for the parts of any product by some heuristic methods as is described in Section 3.2.

The Johnson algorithm to find the sequence of products:

1. Determine the PT_h according to Eqs. (15), (16), (17), and (18) and A_h as shown in the problem for all products.
2. Suppose $U=\{h \quad H|\mathrm{PT}_h<A_h\}$ and $V=\{h\in H|\mathrm{PT}_h\geq A_h\}$.
3. Sort the set of U nondecreasing in PT_h and set of V nonincreasing in A_h.
4. Determine the sequence of products according the set of U and V after that.

3.2 Heuristic methods to determine the sequence and schedule of parts

After determining the sequence of the product, six heuristic methods are used to schedule and assign the parts of any product on the machines at stage 1 and stage 2 in the hybrid flow shop. In this section, also the result of some heuristic algorithms that was introduced by some researchers has been used (for example, [15–17]). The motivation for developing and using heuristic algorithms to find approximate solutions for the considered problem is twofold: Firstly, this problem is strongly NP-hard, and there is no viable optimum solution approach to solve this problem generally and no limit of the number of jobs and machines. Secondly, a quick and good solution is required in practice. Especially in real-time machine vision problems, the tasks should be processed within a short time period. Furthermore, since the machine vision problem itself is computationally intensive, heuristics that will be used by the scheduler of the system are preferred to be less computationally intensive [15]. This has led us to use simple heuristic methods for this problem. These six heuristic methods are presented as follows:

I: Find a sequence S_1 by sorting the parts in nondecreasing sum of process time at stages 1 and 2. $\left(\text{Total process time/TPT} = \sum_{l=1}^{2} P_{lj} \text{ for } j\in J_h\right)$

II: Find a sequence S_2 by sorting the parts in nondecreasing process time at stage 1 of the hybrid flow shop. $\left(P_{1j} \text{ for } j\in J_h\right)$

III: Find a sequence S_3 by sorting the parts in nondecreasing process time at stage 2 of hybrid flow shop. $\left(P_{2j} \text{ for } j\in J_h\right)$

Table 4 Four generated category of test problems

Problem name	Number of jobs	Number of parts	Number of machines at stage 1	Number of machines at stage 2	Process time at stage 1	Process time at stage 2	Assembly time of jobs	Type of problem
AHF1	10	[2, 10]	2	2	[0, 100]	[0, 100]	[100, 300]	PT1
AHF2	10	[2, 10]	3	2	[0, 100]	[0, 100]	[100, 300]	PT1
AHF3	10	[2, 10]	4	2	[0, 100]	[0, 100]	[100, 300]	PT1
AHF4	10	[2, 10]	2	3	[0, 100]	[0, 100]	[100, 300]	PT1
AHF5	10	[2, 10]	2	4	[0, 100]	[0, 100]	[100, 300]	PT1
AHF6	10	[4, 12]	2	2	[0, 100]	[0, 100]	[100, 300]	PT3
AHF7	10	[4, 12]	3	2	[0, 100]	[0, 100]	[100, 300]	PT3
AHF8	10	[4, 12]	4	2	[0, 100]	[0, 100]	[100, 300]	PT3
AHF9	10	[4, 12]	2	3	[0, 100]	[0, 100]	[100, 300]	PT3
AHF10	10	[4, 12]	2	4	[0, 100]	[0, 100]	[100, 300]	PT3
AHF11	10	[6, 16]	2	2	[0, 100]	[0, 100]	[100, 300]	PT2
AHF12	10	[6, 16]	3	2	[0, 100]	[0, 100]	[100, 300]	PT2
AHF13	10	[6, 16]	4	2	[0, 100]	[0, 100]	[100, 300]	PT2
AHF14	10	[6, 16]	2	3	[0, 100]	[0, 100]	[100, 300]	PT2
AHF15	10	[6, 16]	2	4	[0, 100]	[0, 100]	[100, 300]	PT2
AHF16	50	[2, 10]	2	2	[0, 100]	[0, 100]	[100, 300]	PT1
AHF17	50	[2, 10]	3	2	[0, 100]	[0, 100]	[100, 300]	PT1
AHF18	50	[2, 10]	4	2	[0, 100]	[0, 100]	[100, 300]	PT1
AHF19	50	[2, 10]	2	3	[0, 100]	[0, 100]	[100, 300]	PT1
AHF20	50	[2, 10]	2	4	[0, 100]	[0, 100]	[100, 300]	PT1
AHF21	50	[4, 12]	2	2	[0, 100]	[0, 100]	[100, 300]	PT3
AHF22	50	[4, 12]	3	2	[0, 100]	[0, 100]	[100, 300]	PT3
AHF23	50	[4, 12]	4	2	[0, 100]	[0, 100]	[100, 300]	PT3
AHF24	50	[4, 12]	2	3	[0, 100]	[0, 100]	[100, 300]	PT3
AHF25	50	[4, 12]	2	4	[0, 100]	[0, 100]	[100, 300]	PT3
AHF26	50	[6, 16]	2	2	[0, 100]	[0, 100]	[100, 300]	PT2
AHF27	50	[6, 16]	3	2	[0, 100]	[0, 100]	[100, 300]	PT2
AHF28	50	[6, 16]	4	2	[0, 100]	[0, 100]	[100, 300]	PT2
AHF29	50	[6, 16]	2	3	[0, 100]	[0, 100]	[100, 300]	PT2
AHF30	50	[6, 16]	2	4	[0, 100]	[0, 100]	[100, 300]	PT2
AHF31	100	[2, 10]	2	2	[0, 100]	[0, 100]	[100, 300]	PT1
AHF32	100	[2, 10]	3	2	[0, 100]	[0, 100]	[100, 300]	PT1
AHF33	100	[2, 10]	4	2	[0, 100]	[0, 100]	[100, 300]	PT1
AHF34	100	[2, 10]	2	3	[0, 100]	[0, 100]	[100, 300]	PT1
AHF35	100	[2, 10]	2	4	[0, 100]	[0, 100]	[100, 300]	PT1
AHF36	100	[4, 12]	2	2	[0, 100]	[0, 100]	[100, 300]	PT3
AHF37	100	[4, 12]	3	2	[0, 100]	[0, 100]	[100, 300]	PT3
AHF38	100	[4, 12]	4	2	[0, 100]	[0, 100]	[100, 300]	PT3
AHF39	100	[4, 12]	2	3	[0, 100]	[0, 100]	[100, 300]	PT3
AHF40	100	[4, 12]	2	4	[0, 100]	[0, 100]	[100, 300]	PT3
AHF41	100	[6, 16]	2	2	[0, 100]	[0, 100]	[100, 300]	PT2
AHF42	100	[6, 16]	3	2	[0, 100]	[0, 100]	[100, 300]	PT2
AHF43	100	[6, 16]	4	2	[0, 100]	[0, 100]	[100, 300]	PT2
AHF44	100	[6, 16]	2	3	[0, 100]	[0, 100]	[100, 300]	PT2
AHF45	100	[6, 16]	2	4	[0, 100]	[0, 100]	[100, 300]	PT2
AHF46	150	[2, 10]	2	2	[0, 100]	[0, 100]	[100, 300]	PT1
AHF47	150	[2, 10]	3	2	[0, 100]	[0, 100]	[100, 300]	PT1

Table 4 (continued)

Problem name	Number of jobs	Number of parts	Number of machines at stage 1	Number of machines at stage 2	Process time at stage 1	Process time at stage 2	Assembly time of jobs	Type of problem
AHF48	150	[2, 10]	4	2	[0, 100]	[0, 100]	[100, 300]	PT1
AHF49	150	[2, 10]	2	3	[0, 100]	[0, 100]	[100, 300]	PT1
AHF50	150	[2, 10]	2	4	[0, 100]	[0, 100]	[100, 300]	PT1
AHF51	150	[4, 12]	2	2	[0, 100]	[0, 100]	[100, 300]	PT3
AHF52	150	[4, 12]	3	2	[0, 100]	[0, 100]	[100, 300]	PT3
AHF53	150	[4, 12]	4	2	[0, 100]	[0, 100]	[100, 300]	PT3
AHF54	150	[4, 12]	2	3	[0, 100]	[0, 100]	[100, 300]	PT3
AHF55	150	[4, 12]	2	4	[0, 100]	[0, 100]	[100, 300]	PT3
AHF56	150	[6, 16]	2	2	[0, 100]	[0, 100]	[100, 300]	PT2
AHF57	150	[6, 16]	3	2	[0, 100]	[0, 100]	[100, 300]	PT2
AHF58	150	[6, 16]	4	2	[0, 100]	[0, 100]	[100, 300]	PT2
AHF59	150	[6, 16]	2	3	[0, 100]	[0, 100]	[100, 300]	PT2
AHF60	150	[6, 16]	2	4	[0, 100]	[0, 100]	[100, 300]	PT2

IV: Find a sequence S_4 by sorting the parts in nonincreasing sum of process time at stages 1 and 2 of the hybrid flow shop. $\left(\text{Total process time/TPT} = \sum_{l=1}^{2} P_{lj} \text{ for } j \in J_h\right)$

V: Find a sequence S_5 by sorting the parts in nonincreasing process time at stage 1 of the hybrid flow shop. $\left(P_{1j} \text{ for } j \in J_h\right)$

VI: Find a sequence S_6 by sorting the parts in nonincreasing process time at stage 2 of hybrid flow shop. $\left(P_{2j} \text{ for } j \in J_h\right)$

Note 1. The above heuristics are applied for the sequencing of parts for each product (h=1,2,3,…, H).

Note 2. After determining the sequence for the set parts of each product, the parts will be assigned according to their position on the first machine that is available in the second stage.

Note 3. Preemption is allowed between the sets of parts for each product in stage 2 of the hybrid flow shop, but it is not allowed between parts of different products.

By combination of the four algorithms A, B, C, and D to determine the sequence of the product (Section 3.1) and six heuristic algorithm methods I, II, III, IV, V, and VI to schedule the parts of any product on the machines at stage 1 and stage 2 in the hybrid flow shop (Section 3.2), there will be a total of 24 algorithms to schedule the products and their parts as shown in Table 2.

3.3 Propose two lower bounds

The hybrid flow shop scheduling problem, in most cases, is NP-hard [5, 8], so that it is very difficult and time consuming to obtain the optimal solution for HFS as an additional assembly stage. Hence, we introduce and improve two lower bounds for evaluation the heuristic solutions.

It is known that the shortest processing time (SPT) rule minimizes the total completion time in the case of parallel machines [7, 18]. Thus, the SPT rule is going to be adapted on the parallel machines hybrid flow shop stage, so that the total completion time at the second stage can be considered as a lower bound. In other words, the processing time of the pre-assembly stage is not considered, so that the total completion time at the second stage by using the SPT rule is always no larger than the total completion time of any feasible schedule [18].

It is clear that the idle time of producing the first product at the start always occurs in the assembly stage, because the assembly operation cannot begin until pre-assembly processing at the first stage is finished. Therefore, in order to improve the lower bound by considering this idle time, a modified SPT rule is proposed. In other words, the minimum idle time for assembly stage is minimum completion time of parts for any product. So, this minimum time can be computed as minimum parallelization time (MPT):

$$\text{MTP} = \min_{h} \left\{ \max_{j \in J_h} \left(\max\left(\sum \left(P_{1j} + P_{2j}\right)\right), \left(\frac{\sum P_{1j}}{K_1} + \min P_{2j}\right), \left(\frac{\sum P_{2j}}{K_2} + \min P_{1j}\right) \right) \right\} \text{ for } \quad h = 1, 2, \ldots, H \tag{19}$$

Table 5 Problems and results of 24 algorithms

Problem	Algorithm																							
	AI	AII	AIII	AIV	AV	AVI	BI	BII	BIII	BIV	BV	BVI	CI	CII	CIII	CIV	CV	CVI	DI	DII	DIII	DIV	DV	DVI
AHF1	2,297	2,320	2,278	2,276	2,271	2,303	2,332	2,346	2,304	2,293	2,280	2,338	2,241	2,249	2,227	2,226	2,226	2,241	2,250	2,261	2,227	2,223	2,220	2,256
AHF2	2,126	2,136	2,121	2,122	2,117	2,140	2,153	2,156	2,142	2,147	2,147	2,156	2,079	2,085	2,080	2,080	2,080	2,082	2,075	2,078	2,074	2,079	2,077	2,081
AHF3	2,223	2,227	2,214	2,219	2,213	2,239	2,232	2,236	2,227	2,232	2,226	2,240	2,171	2,179	2,172	2,171	2,168	2,176	2,165	2,166	2,166	2,165	2,164	2,163
AHF4	2,235	2,239	2,227	2,203	2,197	2,222	2,225	2,236	2,205	2,190	2,189	2,233	2,193	2,192	2,187	2,167	2,167	2,172	2,185	2,203	2,174	2,166	2,156	2,196
AHF5	2,293	2,300	2,277	2,258	2,252	2,279	2,285	2,289	2,277	2,245	2,246	2,276	2,231	2,228	2,225	2,210	2,210	2,219	2,238	2,244	2,218	2,201	2,199	2,234
AHF6	2,494	2,509	2,472	2,463	2,442	2,520	2,540	2,570	2,498	2,510	2,497	2,578	2,428	2,448	2,402	2,421	2,402	2,456	2,427	2,446	2,394	2,402	2,395	2,445
AHF7	2,391	2,409	2,373	2,390	2,373	2,415	2,463	2,471	2,452	2,454	2,445	2,469	2,380	2,387	2,367	2,367	2,359	2,393	2,379	2,384	2,356	2,357	2,347	2,382
AHF8	2,352	2,370	2,349	2,368	2,346	2,375	2,365	2,378	2,356	2,355	2,345	2,369	2,288	2,284	2,284	2,286	2,279	2,289	2,277	2,279	2,270	2,280	2,267	2,286
AHF9	2,395	2,396	2,354	2,339	2,335	2,382	2,490	2,498	2,471	2,449	2,436	2,485	2,384	2,387	2,348	2,334	2,328	2,373	2,375	2,377	2,332	2,327	2,320	2,373
AHF10	2,332	2,336	2,305	2,278	2,265	2,311	2,356	2,351	2,341	2,290	2,282	2,337	2,273	2,276	2,250	2,219	2,217	2,256	2,273	2,278	2,249	2,223	2,217	2,265
AHF11	3,070	3,118	3,051	3,053	3,027	3,126	3,185	3,246	3,156	3,178	3,146	3,244	3,070	3,118	3,051	3,053	3,027	3,126	3,069	3,118	3,048	3,051	3,026	3,121
AHF12	3,101	3,128	3,082	3,110	3,087	3,140	3,202	3,226	3,175	3,198	3,176	3,232	3,102	3,122	3,086	3,100	3,078	3,129	3,111	3,118	3,099	3,100	3,083	3,129
AHF13	3,062	3,091	3,057	3,081	3,053	3,100	3,129	3,148	3,117	3,131	3,106	3,157	3,071	3,096	3,065	3,084	3,058	3,103	3,085	3,106	3,075	3,079	3,061	3,100
AHF14	2,966	2,980	2,943	2,929	2,901	2,975	3,071	3,085	3,057	3,026	3,004	3,085	2,965	2,973	2,939	2,921	2,895	2,964	2,963	2,970	2,940	2,920	2,897	2,966
AHF15	2,911	2,925	2,894	2,843	2,833	2,903	2,999	2,996	2,966	2,940	2,924	2,987	2,906	2,919	2,893	2,843	2,832	2,899	2,898	2,911	2,885	2,834	2,823	2,892
AHF16	10,123	10,122	10,123	10,117	10,117	10,120	10,139	10,137	10,133	10,125	10,128	10,133	10,119	10,118	10,119	10,118	10,118	10,118	10,118	10,117	10,115	10,114	10,111	10,116
AHF17	10,128	10,137	10,128	10,130	10,127	10,133	10,111	10,111	10,111	10,111	10,111	10,111	10,111	10,111	10,111	10,111	10,111	10,111	10,106	10,107	10,106	10,107	10,106	10,107
AHF18	10,132	10,131	10,133	10,131	10,132	10,132	10,140	10,141	10,137	10,138	10,137	10,140	10,116	10,116	10,116	10,116	10,116	10,116	10,113	10,113	10,113	10,113	10,113	10,113
AHF19	9,955	9,955	9,955	9,953	9,953	9,953	9,973	9,974	9,964	9,959	9,960	9,972	9,948	9,948	9,946	9,945	9,945	9,947	9,948	9,948	9,946	9,945	9,945	9,947
AHF20	10,076	10,076	10,076	10,075	10,075	10,075	10,091	10,093	10,090	10,081	10,082	10,087	10,075	10,075	10,075	10,072	10,072	10,072	10,080	10,080	10,079	10,073	10,073	10,080
AHF21	10,593	10,607	10,558	10,578	10,558	10,615	10,619	10,641	10,607	10,607	10,591	10,654	10,569	10,575	10,545	10,560	10,544	10,588	10,588	10,604	10,562	10,577	10,559	10,610
AHF22	10,399	10,424	10,396	10,425	10,395	10,445	10,363	10,373	10,356	10,363	10,351	10,381	10,271	10,271	10,270	10,261	10,260	10,270	10,270	10,270	10,263	10,259	10,258	10,274
AHF23	10,554	10,565	10,544	10,563	10,544	10,577	10,599	10,608	10,593	10,598	10,588	10,609	10,538	10,539	10,530	10,524	10,523	10,532	10,520	10,524	10,513	10,510	10,508	10,520
AHF24	10,536	10,550	10,521	10,480	10,477	10,525	10,465	10,479	10,443	10,421	10,417	10,458	10,438	10,451	10,418	10,412	10,402	10,430	10,444	10,459	10,423	10,416	10,405	10,443
AHF25	10,442	10,436	10,406	10,382	10,376	10,426	10,380	10,387	10,361	10,332	10,327	10,375	10,337	10,344	10,304	10,295	10,293	10,326	10,334	10,341	10,305	10,294	10,292	10,329
AHF26	14,162	14,208	14,133	14,163	14,131	14,200	14,279	14,335	14,256	14,273	14,234	14,322	14,164	14,211	14,136	14,160	14,124	14,205	14,172	14,212	14,137	14,165	14,130	14,208
AHF27	14,166	14,194	14,138	14,189	14,167	14,207	14,255	14,276	14,239	14,258	14,228	14,295	14,187	14,205	14,165	14,195	14,179	14,211	14,214	14,231	14,196	14,201	14,186	14,226
AHF28	14,140	14,162	14,131	14,158	14,132	14,192	14,156	14,178	14,150	14,167	14,145	14,193	14,156	14,163	14,148	14,150	14,134	14,174	14,159	14,164	14,148	14,146	14,135	14,172
AHF29	13,988	13,993	13,963	13,932	13,924	13,984	14,033	14,038	14,005	13,983	13,967	14,038	13,988	13,993	13,963	13,932	13,924	13,984	13,988	13,993	13,963	13,932	13,924	13,984
AHF30	14,215	14,219	14,186	14,132	14,133	14,182	14,257	14,265	14,230	14,193	14,183	14,245	14,215	14,219	14,186	14,132	14,133	14,182	14,215	14,219	14,186	14,132	14,133	14,182
AHF31	10,442	10,436	10,406	10,382	10,376	10,426	10,380	10,387	10,361	10,332	10,327	10,375	10,337	10,344	10,304	10,295	10,293	10,326	10,334	10,341	10,305	10,294	10,292	10,329
AHF32	14,162	14,208	14,133	14,163	14,131	14,200	14,279	14,335	14,256	14,273	14,234	14,322	14,164	14,211	14,136	14,160	14,124	14,205	14,172	14,212	14,137	14,165	14,130	14,208
AHF33	14,166	14,194	14,138	14,189	14,167	14,207	14,255	14,276	14,239	14,258	14,228	14,295	14,187	14,205	14,165	14,195	14,179	14,211	14,214	14,231	14,196	14,201	14,186	14,226
AHF34	14,140	14,162	14,131	14,158	14,132	14,192	14,156	14,178	14,150	14,167	14,145	14,193	14,156	14,163	14,148	14,150	14,134	14,174	14,159	14,164	14,148	14,146	14,135	14,172
AHF35	13,988	13,993	13,963	13,932	13,924	13,984	14,033	14,038	14,005	13,983	13,967	14,038	13,988	13,993	13,963	13,932	13,924	13,984	13,988	13,993	13,963	13,932	13,924	13,984
AHF36	14,215	14,219	14,186	14,132	14,133	14,182	14,257	14,265	14,230	14,193	14,183	14,245	14,215	14,219	14,186	14,132	14,133	14,182	14,215	14,219	14,186	14,132	14,133	14,182
AHF37	19,756	19,756	19,756	19,754	19,754	19,754	19,752	19,758	19,752	19,748	19,749	19,757	19,754	19,754	19,753	19,752	19,752	19,752	19,750	19,751	19,749	19,747	19,745	19,750
AHF38	19,793	19,793	19,793	19,793	19,793	19,793	19,807	19,810	19,806	19,807	19,806	19,818	19,794	19,794	19,794	19,794	19,794	19,794	19,791	19,791	19,791	19,791	19,791	19,791
AHF39	20,277	20,277	20,277	20,277	20,277	20,277	20,286	20,288	20,287	20,287	20,287	20,288	20,277	20,277	20,277	20,277	20,277	20,277	20,267	20,267	20,267	20,267	20,267	20,267

Table 5 (continued)

Problem	Algorithm																							
	AI	AII	AIII	AIV	AV	AVI	BI	BII	BIII	BIV	BV	BVI	CI	CII	CIII	CIV	CV	CVI	DI	DII	DIII	DIV	DV	DVI
AHF40	19,917	19,917	19,917	19,917	19,917	19,917	19,932	19,933	19,922	19,920	19,920	19,926	19,915	19,915	19,913	19,913	19,913	19,914	19,913	19,918	19,910	19,912	19,910	19,917
AHF41	19,987	19,987	19,987	19,987	19,987	19,987	19,993	19,993	19,987	19,980	19,980	19,990	19,982	19,982	19,981	19,979	19,979	19,982	19,978	19,980	19,975	19,973	19,973	19,980
AHF42	21,124	21,141	21,096	21,108	21,083	21,150	21,092	21,107	21,052	21,065	21,044	21,107	21,097	21,116	21,069	21,079	21,058	21,120	21,100	21,123	21,075	21,090	21,061	21,129
AHF43	20,695	20,720	20,681	20,692	20,673	20,721	20,710	20,719	20,702	20,699	20,696	20,722	20,667	20,672	20,663	20,651	20,654	20,661	20,661	20,671	20,650	20,640	20,637	20,662
AHF44	20,636	20,653	20,629	20,639	20,630	20,664	20,662	20,673	20,659	20,653	20,648	20,668	20,644	20,649	20,642	20,628	20,631	20,642	20,624	20,636	20,623	20,614	20,616	20,630
AHF45	20,398	20,408	20,368	20,343	20,340	20,386	20,392	20,403	20,368	20,339	20,334	20,389	20,324	20,332	20,297	20,289	20,284	20,320	20,318	20,330	20,292	20,287	20,280	20,324
AHF46	20,556	20,563	20,548	20,500	20,495	20,542	20,517	20,514	20,506	20,471	20,473	20,498	20,461	20,463	20,452	20,425	20,427	20,447	20,459	20,464	20,449	20,430	20,424	20,450
AHF47	28,116	28,165	28,073	28,103	28,064	28,175	28,276	28,327	28,252	28,270	28,230	28,319	28,116	28,162	28,075	28,097	28,065	28,171	28,151	28,184	28,115	28,127	28,099	28,187
AHF48	27,440	27,474	27,419	27,469	27,445	27,496	27,451	27,472	27,434	27,437	27,428	27,474	27,457	27,473	27,439	27,448	27,434	27,477	27,475	27,493	27,455	27,458	27,446	27,503
AHF49	27,450	27,487	27,438	27,460	27,433	27,492	27,479	27,482	27,469	27,468	27,459	27,496	27,459	27,468	27,452	27,440	27,438	27,465	27,463	27,478	27,456	27,447	27,441	27,477
AHF50	27,572	27,584	27,542	27,533	27,514	27,582	27,598	27,606	27,572	27,546	27,535	27,590	27,572	27,584	27,543	27,533	27,514	27,582	27,572	27,584	27,543	27,533	27,514	27,582
AHF51	27,493	27,508	27,467	27,435	27,424	27,481	27,494	27,507	27,465	27,428	27,421	27,474	27,493	27,508	27,467	27,435	27,424	27,481	27,493	27,508	27,467	27,435	27,424	27,481
AHF52	30,030	30,032	30,028	30,030	30,028	30,032	30,049	30,052	30,044	30,040	30,038	30,047	30,030	30,032	30,028	30,030	30,028	30,032	30,031	30,036	30,028	30,029	30,028	30,036
AHF53	29,971	29,971	29,971	29,971	29,971	29,971	29,973	29,973	29,971	29,971	29,971	29,971	29,966	29,966	29,966	29,966	29,966	29,966	29,962	29,962	29,961	29,961	29,961	29,961
AHF54	30,177	30,177	30,177	30,177	30,177	30,177	30,187	30,187	30,186	30,189	30,187	30,189	30,175	30,175	30,175	30,175	30,175	30,175	30,169	30,169	30,169	30,169	30,169	30,169
AHF55	30,074	30,074	30,074	30,074	30,074	30,074	30,084	30,085	30,080	30,077	30,076	30,084	30,074	30,074	30,074	30,074	30,074	30,074	30,076	30,077	30,074	30,073	30,073	30,074
AHF56	29,934	29,933	29,933	29,932	29,932	29,933	29,942	29,941	29,939	29,937	29,937	29,940	29,935	29,935	29,933	29,931	29,931	29,934	29,934	29,934	29,932	29,930	29,930	29,933
AHF57	30,882	30,911	30,865	30,864	30,844	30,902	30,777	30,798	30,759	30,762	30,742	30,797	30,833	30,852	30,818	30,829	30,813	30,853	30,857	30,875	30,836	30,846	30,826	30,870
AHF58	30,772	30,791	30,756	30,771	30,757	30,797	30,780	30,787	30,775	30,770	30,767	30,790	30,773	30,780	30,765	30,760	30,755	30,773	30,770	30,781	30,763	30,760	30,752	30,778
AHF59	30,762	30,778	30,761	30,772	30,755	30,794	30,720	30,718	30,717	30,718	30,711	30,719	30,694	30,696	30,697	30,692	30,694	30,697	30,682	30,691	30,683	30,679	30,681	30,691
AHF60	30,654	30,657	30,641	30,618	30,611	30,649	30,599	30,599	30,587	30,565	30,561	30,585	30,596	30,597	30,585	30,566	30,566	30,588	30,600	30,602	30,583	30,566	30,567	30,596

Therefore, the first idle time on the assembly stage is no smaller than the time value MPT even if any product or part comes to the first position of the job sequence on the first stage (hybrid flow shop). Hence, by adding the assembly time for all products, we can calculate the improved lower bound as shown below:

$$\mathrm{LB1} = \min_{h}\left\{\max_{j\in J_h}\left(\max\left(\sum\left(P_{1j}+P_{2j}\right)\right),\left(\frac{\sum P_{1j}}{K_1}+\min P_{2j}\right),\left(\frac{\sum P_{2j}}{K_2}+\min P_{1j}\right)\right)\right\}+\sum_{h=1}^{H}A_h \ \text{for} \quad h=1,2,\ldots,H \tag{20}$$

If the assembly stage is a bottleneck, then the jobs always have idle times to start at the assembly stage (except the parts that belong to the product on the first position), and the assembly stage will be always busy during flow time after completion of all parts of the product on the first position. So, the total completion time of the optimum solution can be close to LB1.

Vice versa, if the hybrid flow shop (as first stage) is the bottleneck stage, then we will have idle times at the assembly stage. In this condition, the first stage will be always busy during flow time from start to before starting assembly operation of the last product. Therefore, the completion time depends on process operation on hybrid flow shop, and it will have a big difference

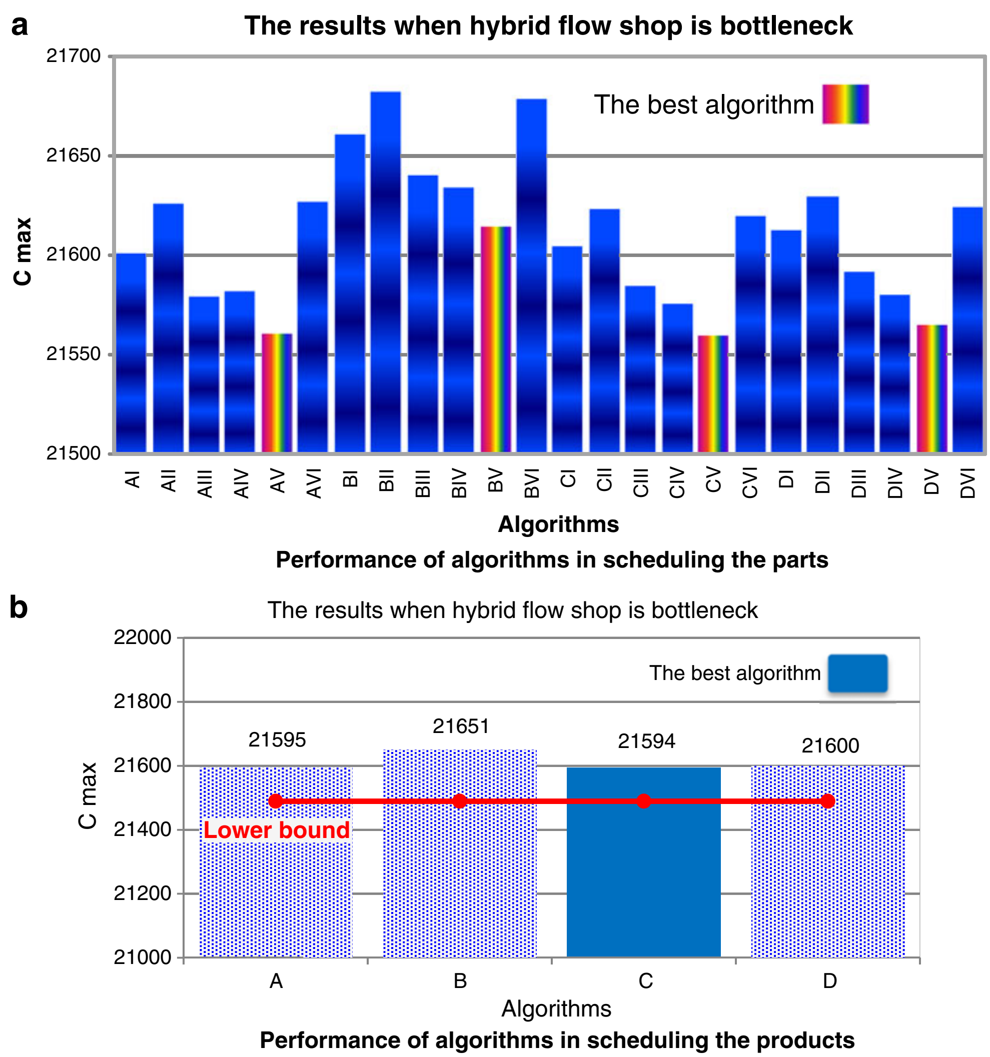

Fig. 5 The result of algorithms in condition that the hybrid flow shop stage is a bottleneck. **a** Performance of algorithms in scheduling the parts. **b** Performance of algorithms in scheduling the products

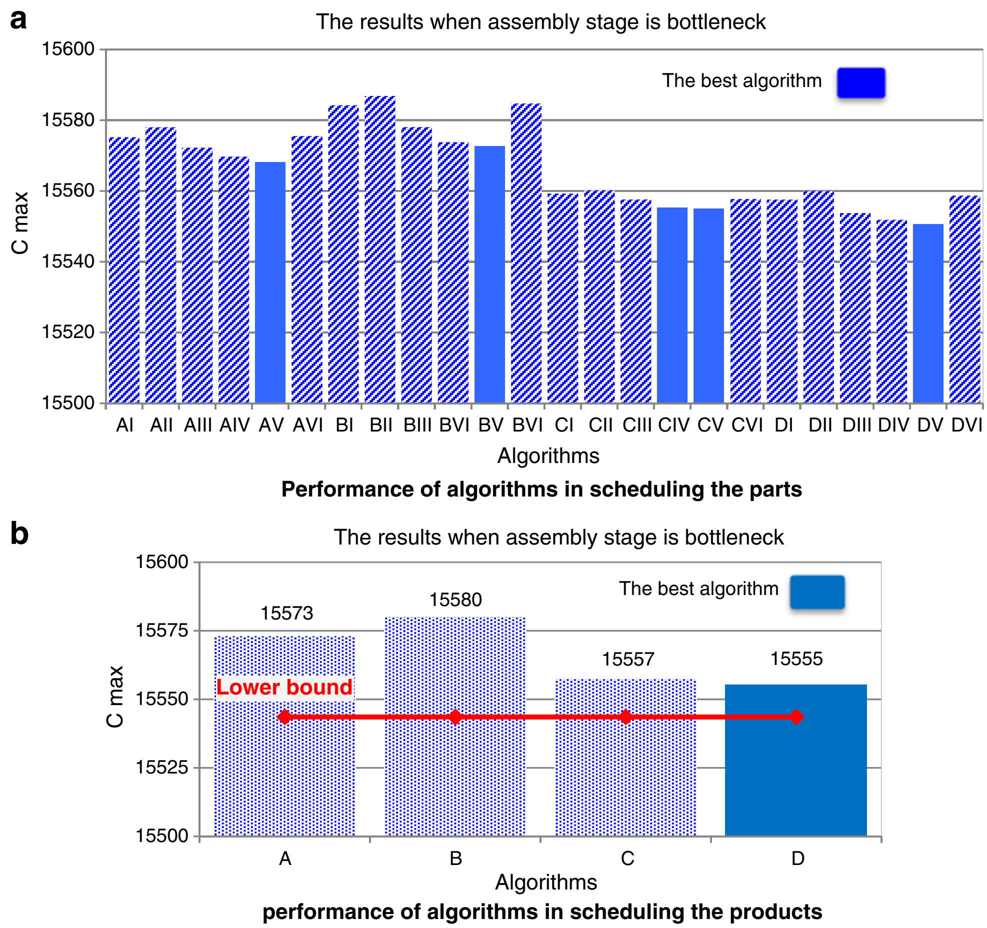

Fig. 6 The result of algorithms in condition that the assembly stage is a bottleneck. **a** Performance of algorithms in scheduling the parts. **b** Performance of algorithms in scheduling the products

from LB1. Hence, we introduce another lower bound as shown below:

$$\mathrm{LB2} = \max\left\{\left(\frac{\sum_{j=1}^{n} P_{1j}}{K_1} + \min P_{2j}\right), \left(\frac{\sum_{j=1}^{n} P_{2j}}{K_2} + \min P_{1j}\right)\right\} + \min A_h \text{ for } \quad j = 1, 2, \ldots, n \quad \text{and} \quad h = 1, 2, \ldots, H \tag{21}$$

Therefore, both LB1 and LB2 are used to cover all conditions and various process times and assembly times.

4 Experimental design

The computational experiments are used to evaluate the performance of the proposed algorithms. We considered three types of problems as shown in Table 3. The type 1 problems (PT1) present the situation that the hybrid flow shop stage is a bottleneck; in type 2 problems (PT2), the assembly stage is the bottleneck, and in type 3 problems (PT3), there is a balanced condition between the two stages. These three types of problems are generated by setting the number of parts. Also, the number of machines at both the two stages and number of jobs (products) are changed at each type of problem to have various problems as shown in Table 3.

Suppose that $[P_\mathrm{L}, P_\mathrm{U}]$ is the range of the number of parts. That is, P_L is the lower limit and P_U is the upper limit of the number of parts. Also, suppose that $[T1_\mathrm{L}, T1_\mathrm{U}]$ is the range of processing time of parts in the first stage of the hybrid flow shop, $[T2_\mathrm{L}, T2_\mathrm{U}]$ is the range of processing time of parts in the second stage of the hybrid flow shop, and $[\mathrm{Ah}_\mathrm{L}, \mathrm{Ah}_\mathrm{U}]$ is the range of assembly time of products. We define index I below to identify the type of the problems:

$$I = \left[H \times \frac{\mathrm{Ah}_\mathrm{L} + \mathrm{Ah}_\mathrm{U}}{2}\right] / \max\left\{\left[H \times \frac{P_\mathrm{L} + P_\mathrm{U}}{2} \times \frac{T1_\mathrm{L} + T1_\mathrm{U}}{2} / K_1\right], \left[H \times \frac{P_\mathrm{L} + P_\mathrm{U}}{2} \times \frac{T2_\mathrm{L} + T2_\mathrm{U}}{2} / K_2\right]\right\}$$

Because the range for number of parts and their processing times and also the assembly time of products are uniform, it is clear that:

If $I<1$, then the hybrid flow shop stage will be the bottleneck.
If $I>1$, then the assembly stage will be the bottleneck.
Else ($I\approx1$), then there will be a balanced condition between the two stages.

Hence, we have generated three types of problems as shown in Table 3.

Also, four categories of problems based on the number of products are considered (note: the number of products does not change the bottleneck). We consider the number of product in a range between 10 and 150 (10, 50, 100, and 150), the range of process time at stages 1 and 2 of the hybrid flow shop between [0, 100] for all parts, and assembly time between [100, 300] for all products. The other data are shown in Table 4.

5 Comparisons of results

This section presents the results of the heuristic methods described in Section 3. The present algorithms are coded in MATLAB 7/10/0/499 (R2010a). The experiments are executed on a Pc with a 2.0-GHz Intel Core 2 Duo processor and 1 GB of RAM memory. Each problem has been run ten times by each algorithm. The average of results obtained of ten runs of each problem is presented in Table 5 for each problem and algorithm. The performances of presented algorithms are shown in Figs. 5, 6, and 7. These figures show that in all conditions, heuristic 5 (nonincreasing process time at stage 1 of the hybrid flow shop) for scheduling the parts has the best result. The results also show that algorithms C and D have the best performance in scheduling of products totally. The results of four algorithms, average of six methods for each algorithm, are shown in panel II of Figs. 5, 6, and 7.

Figure 8 shows the amount of lower bounds (LB1, LB2) when the bottleneck stage is changed. The main amount of LB1 is dependent on assembly time. The assembly time was supposedly fixed in interval [100, 300];therefore, the amount

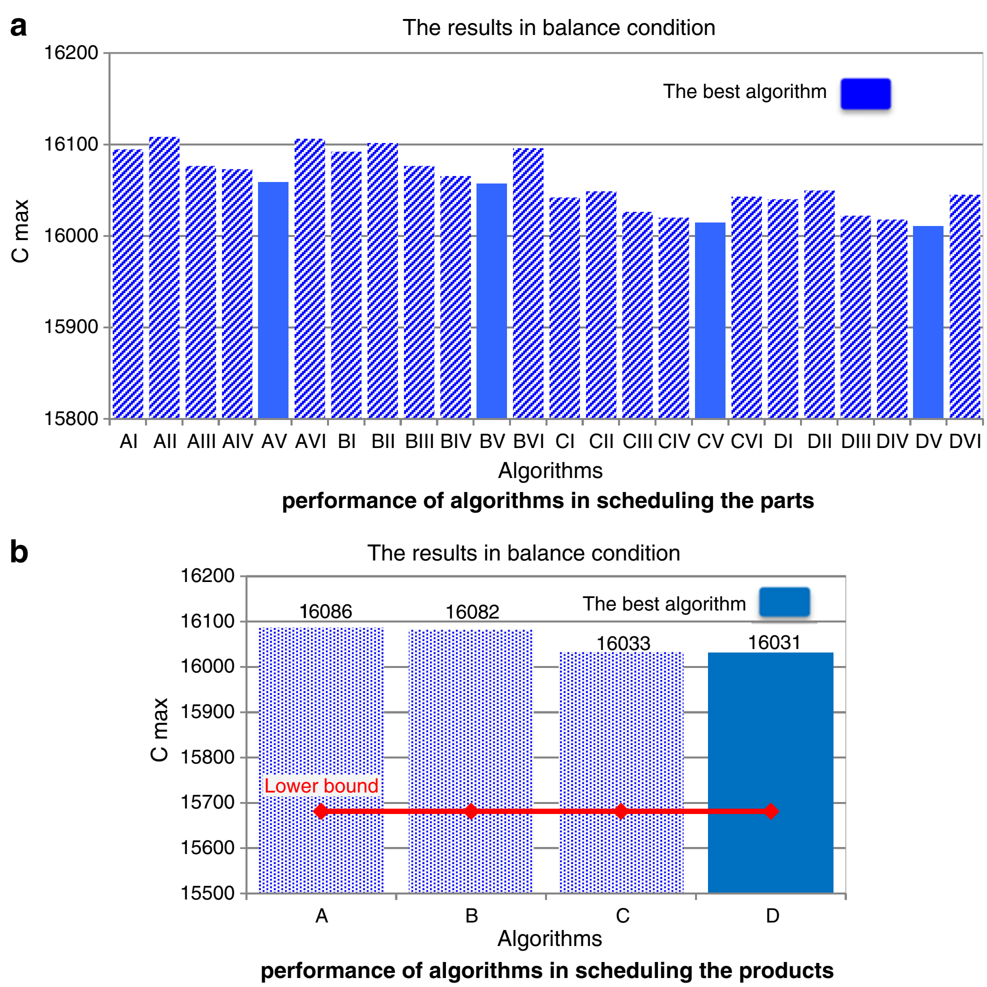

Fig. 7 The result of algorithms in a balanced condition. **a** Performance of algorithms in scheduling the parts. **b** Performance of algorithms in scheduling the products

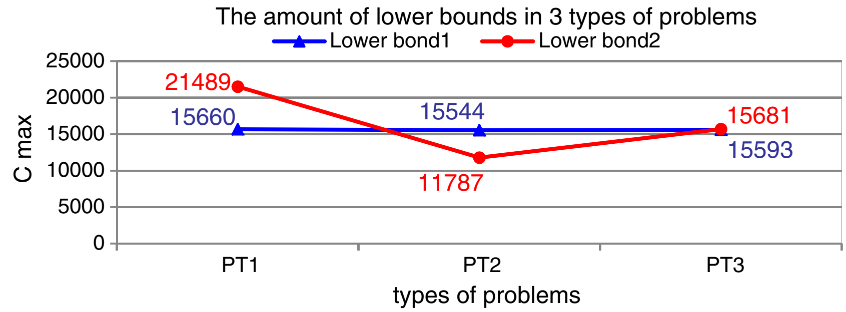

Fig. 8 Variation of lower bounds in three conditions

of LB1 is approximately constant. The main amount of LB2 is dependent on the hybrid flow shop. It is acceptable that when the number of machines in the hybrid flow shop is less and this stage is a bottleneck, then the amount of LB2 is greater. When the number of machines in the hybrid flow shop is more and the assembly stage is a bottleneck (attention that the time of process and assembly is fixed), the amount of LB2 is less than LB1. It is clear that in a balanced condition, the amount of two lower bounds will be approximately equal.

Table 6 shows the summary results of the algorithms and their deviation from lower bounds whichever is more. From Eqs. (20) and (21), it is clear that when the hybrid flow shop is a bottleneck, LB2 gives a greater value, and when the assembly stage is a bottleneck, LB1 gives a greater value. In these two situations, we have lower bounds near the optimum solution because the result of the algorithms are close to the lower bounds. Table 6 shows that when there is no balanced situation, the maximum deviation as a result of the algorithms is 0.75 %, and it happened for algorithm *B*. When there is a balanced situation, none of two lower bounds can present a value near the optimum solution, and in this condition, the amount of lower bounds will be better than the optimum solution in comparison with a situation where is no balanced condition. Also, Table 6 shows that in a balanced condition, the maximum deviation as a result of the algorithms is 2.58 %, and it happened for algorithm *A*.

6 Summary and conclusion

This paper considers a two-stage assembly hybrid flow shop scheduling problem. This production system contains a hybrid flow shop followed by an assembly stage. It was assumed that several products of different kinds are ordered to be produced. Set parts for the products are manufactured in the hybrid flow shop, and products are assembled after the parts are ready in the assembly stage. Four algorithms were extensions based on the Johnson algorithm in order to schedule the products, and six methods are presented to schedule the parts and assign them to machines in each stage of the hybrid flow shop. Moreover two lower bounds were introduced and improved for evaluation of the heuristic solutions. The algorithms were tested in three situations: (1) when the stage of the part fabrication is a bottleneck, (2) when the assembly stage is the bottleneck, and (3) when there is a balanced condition.

The result shows that when the hybrid flow shop is the bottleneck, algorithms *A* and *B* give the best result with 0.49 % deviation from the lower bound. In this condition, lower bound 2 has a greater amount than lower bound 1. When the assembly stage is the bottleneck, algorithms *D* and *C* have just 0.08 and 0.09 % deviation from the lower bound, respectively, and lower bound 1 has a greater amount in this condition. In a balanced condition, algorithms *D* and *C* have the best performance, and their results have 2.23 and 2.24 % deviation from the lower bound, respectively. In this condition, the amounts of two lower bounds are approximately equal, and the results of algorithms have more deviation from the lower bounds.

As a further study, this problem with a number of products of the same kind may be interesting. Also, consideration of the limitation in buffers is suggested. The study of other objective functions such as sum of tardiness, sum of earliness, mean completion time, and so on will be useful in this problem.

Table 6 Summary results of the heuristic algorithms and their deviation from lower bound

Algorithm	Type 1 problems (PT1)		Type 2 problems (PT2)		Type 3 problems (PT4)	
	Result of algorithm	Deviation (%)	Result of algorithm	Deviation (%)	Result of algorithm	Deviation (%)
A	21,595	0.49	15,573	0.19	16,086	2.58
B	21,651	0.75	15,580	0.23	16,082	2.55
C	21,594	0.49	15,557	0.09	16,033	2.24
D	21,600	0.51	15,555	0.08	16,031	2.23

References

1. Lee CY, Cheng TCE, Lin BMT (1993) Minimizing the makespan in the 3-machine assembly-type flowshop scheduling problem. Manag Sci 39:616–625
2. Allahverdi A, Al-Anzi FS (2009) The two-stage assembly scheduling problem to minimize total completion time with setup times. Comput Oper Res 36:2740–2747
3. Potts CN, Sevast'Janov SV, Strusevich VA, Van Wassenhove LN, Zwaneveld CM (1995) The two-stage assembly scheduling problem: complexity and approximation. Oper Res 43:346–355
4. Koulamas Ch, Kyparisis GJ (2001) The three-stage assembly flow shop scheduling problem. Comput Oper Res 28:689–704
5. Blazewicz J, Ecker K, Pesch E, Schmidt G, Weglarz J (2007) Handbook on scheduling from theory to application. Springer, USA
6. Quadt D, Kuhn H (2007) A taxonomy of flexible flow line scheduling procedures. Eur J Oper Res 178:686–698
7. Pinedo ML (2008) Scheduling theory, algorithms, and systems, 3rd edn. Springer, USA
8. Ruiz R, Vazquez-Rodriguez JA (2010) Invited review the hybrid flow shop scheduling problem. Eur J Oper Res 205:1–18
9. Hariri AMA, Potts CN (1997) A branch and bound algorithm for the two-stage assembly scheduling problem. Eur J Oper Res 103:547–556
10. Cheng TCE, Wang G (1999) Scheduling the fabrication and assembly of components in a two-machine flow shop. IIE Trans 31:135–143
11. Yokoyama M (2001) Hybrid flow-shop scheduling with assembly operations. Int J Prod Econ 73:103–116
12. Sun X, Morizawa K, Nagasawa H (2003) Powerful heuristics to minimize makespan in fixed, 3-machine, assembly-type flow shop scheduling. Eur J Oper Res 146:498–516
13. Yokoyama M, Santos DL (2005) Three-stage flow-shop scheduling with assembly operations to minimize the weighted sum of product completion times. Eur J Oper Res 161:754–770
14. Al-Anzi FS, Allahverdi A (2009) Heuristics for a two-stage assembly flow shop with bicriteria of maximum lateness and makespan. Comput Oper Res 36:2682–2689
15. Oguz C, Ercan MF, Cheng TCE, Fung YF (2003) Heuristic algorithms for multiprocessor task scheduling in a two-stage hybrid flow-shop. Eur J Oper Res 149:390–403
16. Shaukat A, Brah LLL (1999) Heuristics for scheduling in a flow shop with multiple processors. Eur J Oper Res 113:113–122
17. Gupta JND, Kruger K, Lauff V, Werner F, Sotskov YN (2002) Heuristics for hybrid flow shops with controllable processing times and assignable due dates. Comput Oper Res 29:1417–1439
18. Sung CS, Ah KH (2008) A two-stage multiple-machine assembly scheduling problem for minimizing sum of completion times. Int J Prod Econ 113:1038–1048