International Journal of Industrial Engineering & Production Research (2017)

March 2017, Volume 28, Number 1
pp. 61- 74

DOI: 10.22068/ijiepr.28.1.61 http://IJIEPR.iust.ac.ir/

A Heuristic Approach to Solve Hybrid Flow-Shop Scheduling Problem with Unrelated Parallel Machines

Ebrahim Asadi Gangraj

Ebrahim Asadi Gangraj, *Babol Noshirvani University of Technology*

KEYWORDS

Bottleneck-based Heuristic,
Lower bound,
Hybrid flow shop,
Makespan.

ABSTRACT

In hybrid flow-shop scheduling problem (HFS) with unrelated parallel machines, a set of n jobs is processed on k machines. A mixed integer linear programming (MILP) model for the HFS scheduling problems with unrelated parallel machines has been proposed to minimize the maximum completion time (makespan). Since the problem is shown to be NP-complete, it is necessary to use heuristic methods to tackle the moderate- to large-scale problems. This article presents a new bottleneck-based heuristic to solve the problem. To improve the performance of the heuristic method, a local search approach is embedded in the structure of the heuristic method. To evaluate the performance of the proposed heuristic method, a new lower bound is developed based on Kurz and Askin [1] lower bound. For evaluation purposes, two series of test problems, small- and large-sized problems, are generated under different production scenarios. The empirical results show that the average differences between lower bound and optimal solution as well as lower bound and heuristic method are equal to 2.56% and 5.23%, respectively. For more investigation, the proposed heuristic method is compared with other well-known heuristics in the literature. The results verify the efficiency of the proposed heuristic method in terms of the number of best solutions.

1. Introduction

In the classical flow shop problem, a set of jobs must be processed on a number of sequential machines, processing routes of all jobs are the same [2]; but in the hybrid flow-shop (HFS) scheduling problem with unrelated parallel machines, there exist one or more unrelated parallel machines in each stage [3]. In scheduling literature, this problem is recognized as a flexible flow shop (FFS), flexible flow line (FFL), hybrid flow line (HFL), or flow shop with multiple processors (FSMP). The hybrid flow shops exist in many real-world manufacturing environments such as semiconductor manufacturing [4], automobile assembly plant [5], packaging industry [6], steel manufacturing [7], printed circuit-board assembly [8], and metal forming

* Corresponding author: ***Ebrahim Asadi Gangraj***

Email: e.asadi@nit.ac.ir

Received 27 February 2017; revised 15 April 2017; accepted 11 June 2017

[9]. Some authors addressed this classical problem in their studies [10-18].

In the HFS with unrelated parallel machines, the job processing times in each stage are different and dependent on the machine type. This may be due to the differences between the machines themselves, to the fact that each type of machine is better suited for a particular job, whereas others are not, or because the job has some particular characteristics and it must be assigned to machines that are physically near to it [3].

In the last decade, there are some comprehensive reviews on hybrid flow-shop scheduling problem. Vignier, Billaut, and Proust [19] accomplished a review of this problem. After that, Kis and Pesch [20] reviewed the exact methods for the k-stage HFS problem with identical parallel machines to minimize makespan or total flow time. Quadt and Kuhnt[16] also proposed taxonomy for k-stage HFS scheduling procedures focusing on heuristic procedures. Finally, Ribas et al. [3] classified papers in the HFS environment from various aspects, such as the different constraints, machine features, solution approaches, and optimization criteria.

The bottleneck phenomena occur frequently in some production systems. In TOC (theory of constraints) production philosophy, Goldratt and Cox [21] stated the idea that the bottleneck resources govern the system's performance. Drum-Buffer-Rope (DBR) proposed by Goldratt and Fox [22] is a popular scheduling approach in bottleneck-based environment. DBR focuses on scheduling bottleneck workstations that affect the upstream and downstream workstations by scheduling on the bottleneck. Therefore, the Bottleneck-based scheduling approach has attracted some researchers in the last decades.

Acero-Dominguez and Paternina-Arboleda [23] proposed a bottleneck-based algorithm for the HFS scheduling problem with related parallel machines and makespan objective function. They proposed a three-step approach: 1) bottleneck identification; 2) computation of release and tail times of each job for the bottleneck stage and scheduling of jobs at the bottleneck stage; 3) scheduling jobs on non-bottleneck resources based on scheduling on the bottleneck. Chen and Chen [24] proposed a three-step bottleneck-based heuristic to solve a flexible flow-line scheduling problem with a bottleneck stage, with the objective of minimizing the makespan.

Paternina-Arboleda Montoya-Torres, Acero-Dominguez, and Herrera-Hernandez [25] dealt with the problem of the makespan minimization on a FFS with *k* stages so that there are one or more identical machines at any stage. They proposed three-step TOC-based heuristic for this problem. With respect to the absolute makespan value, they compared the proposed heuristic with shifting bottleneck heuristic, hybrid shifting bottleneck-local search heuristic [26]. Chen and Chen [27] considered the total tardiness minimization on the FFS problem with unrelated parallel machines in each stage and with a bottleneck stage in the production line. Two bottleneck-based heuristics with three machine selection rules were proposed for the research problem.

Because of NP-completeness of the HFS problem in a strong sense, it is difficult to evaluate the performance of the sub-optimal procedures. Some researchers derived lower bounds for the HFS scheduling problem. Haouari and M'Hallah [28] presented a new lower bound for makespan at two-stage flexible flow shop. Santos, Hunsucker and Deal [29] developed a lower bound for multiple identical FFS problem. The procedure for developing a global bound involves determining a lower bound for each stage. Soewandi and Elmaghrby [30] proposed some lower bounds for the three-stage FFS problem with identical parallel machines based on an auxiliary problem from the original problem. Kurz and Askin [1] improved the lower bound proposed by Leon and Ramamoorthy [31] for the HFS problem with identical parallel machines.

In this research, we consider the HFS scheduling problem with unrelated parallel machines. In the last decades, many researchers considered the HFS problem with identical machines in each stage and the HFS with unrelated parallel machines was rarely studied. Therefore, this paper addresses a new heuristic algorithm for the HFS scheduling problem with unrelated parallel machines based on TOC production philosophy. Also, a new lower bound was proposed based on the lower bound of Kurz and Askin [1] for the research problem to evaluate the heuristic algorithm.

The rest of the paper is organized as follows: in section 2, a mathematical model for the HFS scheduling problem with unrelated parallel machines is described. A bottleneck-based heuristic algorithm is proposed in section 3. Section 4 is dedicated to present a lower bound for the research problem. In section 5, the experimental study is presented to evaluate the lower bound and the proposed heuristic according to some experimental factors. Finally, the main findings of this paper as well as suggestions for the future research areas are addressed in section 6.

2. Model Development

This section is devoted to present a new mixed integer linear programming (MILP) model for the HFS with unrelated parallel machines at each stage. At first, we define the necessary notations to present the MILP model.

Set and index:

j, l : Job index;
i, h: Stage index;
k: Machines index in each stage;
n: Number of jobs;
d: Number of stages;
M_i: Number of machines in stage I;
H: Set of job indices necessary to define decision variables ($H = \{j, l \mid j < l\}$).

Input and parameters:

M: A large number;
d_{ijk} : Processing time of job j in stage i on machine k;

Decision variables:

C_{ij}: Completion time of job j in stage I;
ST_{ij}: Start time of job j in stage I;

$$X_{ijk} = \begin{cases} 1\ if\ job\ j\ is\ processed\ on\ machine\ k\ at\ stage\ i \\ 0 \quad otherwise \end{cases}$$

$$Z_{ilj} = \begin{cases} 1\ if\ job\ l\ is\ processed\ before\ job\ j\ at\ stage\ i \\ 0 \quad otherwise \end{cases}$$

Set H is defined to omit unnecessary decision variables and constraints in the mathematical model expression. If the decision variables are defined regarding the range of their indices, the number of decision variables increases to a huge number and the efficiency of the mathematical model will decrease drastically [32]. For instance, $Y_{ilj}$is equal to 1, the value of Y_{ilj} is 0, and vice versa.

2-1. Model formulation

With respect to the notations summarized in the pervious section, the HFS mathematical model with unrelated parallel machines can be modeled as MILP as follows:

$$Min\ C_{max} \tag{1}$$

subject to:

$$C_{max} \geq ST_{dj} + \sum_{k=1}^{M_d} d_{djk}.X_{djk} \qquad j = 1,2,\ldots,n \tag{2}$$

$$\sum_{k=1}^{M_i} X_{ijk} = 1 \qquad j = 1,2,\ldots,n \& i = 1,2,\ldots,d \tag{3}$$

$$ST_{1j} = 0 \qquad j = 1,2,\ldots,n \tag{4}$$

$$ST_{ij} \geq ST_{i-1,j+} \sum_{k=1}^{M_i} d_{i-1,jk}.X_{i-1,jk} \qquad j = 1,2,\ldots,n \& i = 2,\ldots,d \tag{5}$$

$$ST_{ij} + M(3 - X_{ijk} - X_{ilk} - Z_{ilj}) \geq ST_{il} + \sum_{k=1}^{M_i} d_{ilk}.X_{ilk} \qquad j,l = 1,2,\ldots,n \& (j,l) \in H,\ i = 1,2,\ldots,d \& k = 1,2,\ldots,M_i \tag{6}$$

$$ST_{il} + M(2 - X_{ijk} - X_{ilk} + Z_{ilj}) \geq ST_{ij} + \sum_{k=1}^{M_i} d_{ijk}.X_{ijk} \qquad j,l = 1,2,\ldots,n \& (j,l) \in H,\ i = 1,2,\ldots,d \& k = 1,2,\ldots,M_i \tag{7}$$

$$X_{ijk} \in \{0,1\} \qquad j = 1,2,\ldots,n,\ i = 1,2,\ldots,d \& k = 1,2,\ldots,s_i \tag{8}$$

$$Z_{ilj} \in \{0,1\} \qquad j,l = 1,2,\ldots,n \& (j,l) \in H,\ i = 1,2,\ldots,d \tag{9}$$

The objective function, as presented in Eq. (1), minimizes makespan, and constraint set (2) shows the relation between starting time of each job in the last stage and makespan objective function. Constraint set (3) shows that each job must be assigned to one machine in each stage. Constraint set (4) demonstrates that each job is available at the beginning of the planning horizon. The relationship between starting times for job j in two consecutive stages is stated in the constraint set (5). Constraint sets (6) and (7) show the relationship between two consecutive jobs that are processed on one machine in each stage. If the two jobs, l and j, are processed on one machine and job l is processed before job j, constraint (6) is activated to prevent interference between the processing operations of these two jobs; on the other side, the roles of these two constraint sets are changed. Finally, both constraints (8), (9) force variables X_{ijk} and Z_{ilj} to assume binary values 0 or 1.

3. The Proposed Heuristic Algorithm

This section is dedicated to present the proposed bottleneck-based heuristic for the HFS problem. With respect to unrelated parallel machines in each stage, a machine selection rule will be applied to determine the assignment of the jobs in each stage. This algorithm is based on TOC philosophy and a local search is inserted into the structure of the proposed heuristic to improve the overall performance.

3-1. Assumption

The hybrid flow shop problem, considered in this research, assumes that there are d stages and includes a bottleneck stage B. There are M_i unrelated parallel machines in stage i. There are n jobs with the same routing and must visit all the stages consecutively. The processing time of a job in a stage is known and depends on the machine assigned to the stage. A machine can process only one job at a time and a job must be processed on one machine in any stage. Job preemption is not allowed, and there is no machine breakdown and no set-up time required before the jobs are processed on any machine. In this research, the workload is determined in each stage to identify the bottleneck stage. The workload of stage i is computed by the sum of the processing time at a particular stage divided by its number of machines, denoted as $FR_i = \frac{\sum_{j=1}^{n}\sum_{k=1}^{M_i} d_{ijk}}{M_i}$. The stage with the largest FR_i is defined as the bottleneck stage.

3-2. Heuristic algorithm

This algorithm is based on the theory of constraints (TOC) manufacturing philosophy that state the idea that "bottleneck resources govern the overall system performance" (Goldratt and Fox, 2004). Performance enhancement of the bottleneck stage can improve the overall performance; in other words, finding the bottleneck stage and exploiting it can optimize the whole system performance. Based on this idea, the heuristic algorithm consists of four steps. In the first step of the proposed heuristic approach, the bottleneck stage is identified based on the relative workload in each stage. In the second step, sequence of the jobs at the bottleneck stage is determined based on two proposed parameters: the first one calculates the necessary time for taking the job to bottleneck stage and the second one estimates the minimum time to pass from the bottleneck stage to the last stage; the jobs are arranged by the ascending order of these parameters. At the third step, it can try to sequence the jobs in the non-bottleneck stages: upstream and downstream stages. For this purpose and in the upstream stages, the jobs are arranged based on the completion times in the previous stage and start time in the bottleneck stage. Also, for downstream stages, the jobs are sequenced based on an ascending order of the completion times in the previous stage and estimated time to pass the production line. Finally, after determining the initial sequence, a local search approach is applied to improve the quality of the generated solutions. The algorithm is now described in more details as follows:

Step 1. Finding the bottleneck stage

- For each stage i, compute flow ratio $FR_i = \frac{\sum_{j=1}^{n}\sum_{k=1}^{M_i} d_{ijk}}{M_i}, i = 1,2,\dots,m$. The stage with the largest FR_i is defined as the bottleneck stage and is denoted by B.

Step 2. Sequencing in bottleneck stage B

- For each job j, compute $R_j = \sum_{i=1}^{B-1} \min_k\{d_{ijk}\}, k = 1,2,\dots,M_i, j = 1,2,\dots,n$. R_j denotes the minimum necessary time for job j to get to the bottleneck stage.
- For each job j, compute $D_j = \sum_{i=1}^{m} FR_i - R_j, k = 1,2,\dots,M_i, j = 1,2,\dots,n$. D_j denotes the estimated time needed for job j to pass through from bottleneck stage to the last stage.
- Arrange jobs by increasing order of R_j. If there is more than one job with the same R_j, rank them in increasing order of D_j. If there is more than one job with the same D_j, select job with the largest processing time in the bottleneck stage.

Step 3. Sequencing in non-bottleneck stages

- Upstream stages:
  - For the first stage, $i = 1$, compute for each job j, $\bar{d}_j = \min_k\{d_{1jk}\}, k = 1,2,\dots,s_1$. Arrange jobs by increasing the order of $\bar{d}_j$. If there is more than one job with the same $\bar{p}_j$, select a job with the largest processing time at the bottleneck stage.
  - For upstream stage i, $i = 2,\dots,B$, compute $C_{i-1,j}$ and ST_{Bj}. Arrange jobs by increasing the order of $C_{i-1,j}$. If there is more than one job with the same $C_{i-1,j}$, rank them in an increasing order of ST_{Bj}. If there is more than one job with the same ST_{Bj}, select job with the largest processing time in the bottleneck stage.
- Downstream stages:
  - For downstream stage h, compute $C_{h-1,j}$ and $D_j = \sum_{i=1}^{m} FR_i - R_j, k = 1,2,\dots,s_i, j = 1,2,\dots,n$. Arrange jobs by increasing order of $C_{h-1,j}$. If there is more than one job with the same $C_{h-1,j}$, rank them in increasing order of D_j. If there is more than one job with the same D_j, select job with the largest processing time at the bottleneck stage.

Step 4. Local search

I. Select the first job in the initial sequence and let it be the current partial sequence.

II. Select the next job in the initial sequence and insert the job into the positions before, between, and after every two consecutive jobs of the current partial sequence.

III. Calculate makespan for each partial sequence produced in Step II while adjusting jobs' entering sequence at the bottleneck stage to be the same as that at the first stage.

IV. Select the partial sequence with minimum makespan and let the partial sequence be the current partial sequence.

V. If the current partial schedule includes the entire n jobs, then stop; otherwise, go to Step II.

3-3. Machine selection rule

In order to consider unrelated parallel machines in each stage, we must apply machine selection rule to assign each job to machines. In this research, for a given job, the job is assigned to all the machines (available and unavailable) in each stage and selects a machine that has the earliest completion time. This proposed selection rule may cause idle period of the job. However, due to unrelated parallel machines, an unavailable but more efficient machine may produce an earlier completion time for a given job; in other words, this rule may prefer an unavailable machine with a short processing time to an available machine with a long processing time.

4. Lower Bound

Due to the NP-completeness of the HFS problem with unrelated parallel machines in each stage, it is very hard and time-consuming to obtain the optimum solution for medium- and large-sized problems. On the other hand, we need criteria to evaluate the proposed heuristic. Therefore, we will develop a lower bound to compare the results of the proposed heuristic with a lower bound, so that we can assess the quality of the solutions. This lower bound is based on Kurz and Askin (2001) lower bound which is developed for HFS problem with identical parallel machines. For this purpose, two changes have been implemented in the structure of the lower bound; at first, with respect to unrelated parallel machines at each stage, we rewrite the lower bound based in this new assumption. Subsequently, a new term is applied in the structure of the lower bound to improve its quality. The lower bound, proposed by Kurz and Askin (2001), is as follows:

$$LB = \max_{i=1,2,\dots,d}\left\{\min_j \sum_{s=1}^{i-1} d_{sj} + \sum_{j=1}^{n} d_{ij}\Big/M_i + \min_j \sum_{s=i+1}^{d} d_{sj} + \frac{1}{M_i}\sum_{k=1}^{M_i-1}\left(\min_{j}{}_k \sum_{s=1}^{r-1} d_{sj} - \min_j \sum_{s=1}^{r-1} d_{sj}\right)\right\} \quad (10)$$

The first term states the minimum necessary time to receive the first job to stage i, the second term shows average workload in stage i, the third part indicates the minimum necessary time to receive the last job to last stage, and the last term shows an estimation about necessary time for other jobs from the first stage to stage i.

As mentioned above, in the research problem, we consider unrelated parallel machines at each stage. Thus, we appropriately modify the proposed lower bound based on this assumption as follows:

$$LB = \max_{i=1,2,\dots,d}\left\{\min_j \sum_{s=1}^{i-1} \min_k d_{sjk} + \sum_{j=1}^{n}\sum_{k=1}^{M_i} d_{ijk}\Big/M_i + \min_j \sum_{s=i+1}^{d} \min_k d_{sjk} + \frac{1}{M_i}\sum_{k=1}^{M_i-1}\left(\min_j{}_k \sum_{s=1}^{r-1} \min_k d_{sjk} - \min_j \sum_{s=1}^{r-1} \min_k d_{sjk}\right)\right\} \quad (11)$$

If at a particular stage s, the number of machine in stage s is greater than the number of machine at stage s-1, another job is added to the jobs which are processed at the beginning of the sequence. Thus, the term $\sum_{o=1}^{s-1} min_j\{min_k\, d_{ojk}\} \times (M_i - M_{i-1})^+$ is applied to strengthen the lower bound, as follows:

$$LB = max_{i=1,2,\dots,d}\left\{min_j \sum_{s=1}^{i-1} min_k\, d_{sjk} + \sum_{j=1}^{n}\sum_{k=1}^{M_i} d_{ijk}/M_i + min_j \sum_{s=i+1}^{d} min_k\, d_{sjk} + \frac{1}{M_i}\sum_{k=1}^{M_i-1}\left(\min_j{}_k \sum_{s=1}^{r-1} min_k\, d_{sjk} - \min_j \sum_{s=1}^{r-1} min_k\, d_{sjk}\right) + \sum_{o=1}^{s-1} min_j\{min_k\, d_{ojk}\} \times (M_i - M_{i-1})^+\right\} \quad (12)$$

5. Computational Study

This section is devoted to evaluate the performance of the lower bound and proposed heuristic. For evaluation purposes, two series of test problems are generated under different production scenarios. The first series is dedicated to small-sized problems and the second series is devoted to medium- and large-sized problems.

5-1. Comparison of the proposed lower bound with the optimal solution for small-sized problems

Firstly, to examine the performance of the proposed lower bound, this section compares the lower bound with the optimal solution. We try to find the optimal value of the objective function for test problems using the proposed MILP in this research. Lingo (version 16) is applied to solve the problems, optimally. Due to the NP-completeness of the HFS scheduling problem, it is very expensive to achieve the optimal solutions to the medium- and large-sized problems. Hence, test problems in this section are limited to the small-sized problems. Table 1 summarizes the experimental factors used to define the production scenarios for the 20 small-sized problems.

Tab. 1. Experimental factor for small size problems

Experimental factor	Feature
Number of jobs	U[3,5]
Number of stages	U[2,4]
Number of machines in each stage	U[1,3]
Processing time	U[5,10]

Regarding Table 1, four experimental factors are used to define small-sized problems: number of jobs, number of stages, number of machines in each stage, and processing time. The results of the comparison between lower bound and optimal solution are described in Table 2:

Tab. 2. Lower bound performance evaluation

Test problem	No. of jobs	No. of stages	No. of machines in each stage	Lower bound	Optimal solution	Optimal Gap
1	4	3	2 2 3	26	29*	---
2	4	3	2 1 3	37	39	5.41%
3	5	4	2 2 1 2	53	53	0.00%
4	3	4	2 2 2 1	37	38	2.70%
5	4	3	3 2 2	26	29*	---
6	5	2	2 2	22	26*	---
7	3	3	2 2 3	24	25	4.17%
8	4	2	1 3	27	27	0.00%
9	5	3	1 2 2	55	55	0.00%
10	4	3	2 1 3	46	47	2.17%
11	4	3	2 2 2	25	25	0.00%
12	3	3	3 3 2	24	26	7.69%
13	4	3	2 2 1	42	42	0.00%

14	4	4	3 1 2 2	50	52	4.00%
15	5	2	3 1	53	53	0.00%
16	3	3	3 2 2	23	25	8.70%
17	4	2	2 2	21	22	4.76%
18	4	2	1 2	29	29	0.00%
19	3	3	2 3 3	25	26	4.00%
20	4	4	1 1 2 2	50	50	0.00%

According to Table 2, the proposed lower bound can obtain the optimal solutions in 53% of test problems and solver cannot attain the optimal solution after 10 minutes in three-test problems. The average difference between lower bound and optimal solution (optimal gap) equals 2.56%. Furthermore, in 14 (out of 17) test problems, the optimal gap is lower than 5%. Therefore, we can conclude that the lower bound has good quality to achieve the optimal solutions.

5-2. Comparison of the proposed heuristic with lower bound for medium- and large-sized problems

In order to evaluate the performance of the proposed heuristic method, relative percentage deviation (RPD) from the lower bound is applied as follows:

$$\mathrm{RPD} = \frac{\mathrm{C_{heuristic}} - \mathrm{C_{LB}}}{\mathrm{C_{LB}}} \times 100 \qquad (13)$$

In this equation, $C_{heuristic}$ and C_{LB} are the makespans obtained by the proposed heuristic algorithm and lower bound, respectively. A series of computational experiments are produced to evaluate the proposed heuristic based on different production scenarios. These scenarios consist of five experimental factors: number of jobs, number of stages, number of machines in each stage, variation of processing time, and bottleneck position. Table 3 summarizes the experimental factors used to define production scenarios. The number of jobs has three levels with values set at 10, 20, and 50 (low, medium, and high). The number of stages also has three levels with values set at 3, 6, and 12 (low, medium and high), where the number of machines in each stage is generated from discrete uniform distribution in the range of [2, 4] and [4, 6] (low and high). Processing time has one level in which the non-bottleneck stages have uniform distribution in the range of [5, 10] and bottleneck stage has uniform distribution in the range of [11, 15]. Finally, the bottleneck is located in three places, first 1/3, second 1/3 and third 1/3.

Tab. 3. Experimental factor for medium- and large-sizeی problems

Experimental factor	Levels	Mark
Number of jobs	10 20 50	L M H
Number of stages	3 6 12	L M H
Number of machines in each stage	U[2,4] U[4,6]	L H
Processing time	U[5,10] U[11,15]	Non-bottleneck Bottleneck
Bottleneck position	First 1/3 Second 1/3 Third 1/3	1 2 3

Regarding the five experimental factors considered at Table 3, there are 54 production scenarios and 10 test problems are generated for each scenario. These test problems (540 test problems) are used to compare the performance of the proposed heuristic with that of the lower bound. As mentioned above, the objective function for all the test problems is to minimize the makespan. The experimental results of 540 test problems are provided in the appendix in table 6.

Table 4 shows the average of relative deviation from the lower bound based on some test problems. These results are presented for the various experimental factors. As it can be seen in table 4, in all the instances, the proposed heuristic, in average, has 5.4% difference from the lower bound.

Tab. 4. Average of relative deviation from lower bound (in percent)

Jobs	Bottleneck position	Number of stage					
		L			H		
		3	6	12	3	6	12
10	First 1/3	3.81	2.33	4.15	13.33	12.17	8.66
	Second 1/3	3.34	3.68	4.56	14.15	12.45	8.84
	Third 1/3	3.91	4.15	4.66	16.43	12.94	10.21
20	First 1/3	2.32	3.03	3.32	6.04	5.39	4.55
	Second 1/3	2.48	3.12	3.18	7.16	6.73	5.83
	Third 1/3	5.25	3.23	4.02	6.92	7	6.98
50	First 1/3	1.54	1.6	1.97	3.69	4.18	4.14
	Second 1/3	1.48	1.52	1.71	3.7	4.45	4.37
	Third 1/3	1.64	1.89	1.87	3.32	4.23	5.11

With respect to Table 4, the average RPD for the entire test problems is equal to 5.23%. Furthermore, the computational results show several noteworthy points. When the number of jobs increases, the RPD significantly decreases in any experimental factors. Also, as can be seen, bottleneck position does not affect the RPD in entire scenarios. When the number of stages increases from L to H, the RPD increases significantly. But, in the number of machines in each stage, we cannot identify any distinct pattern.

Furthermore, we applied Figs. 1-3, which present the RPD trends for the number of jobs to confirm the pervious conclusions. In all the figures, each plot name consists of three sections: the first section presents the number of stages, the second section denotes the number of machines in each stage, and the third term identifies the number of jobs.

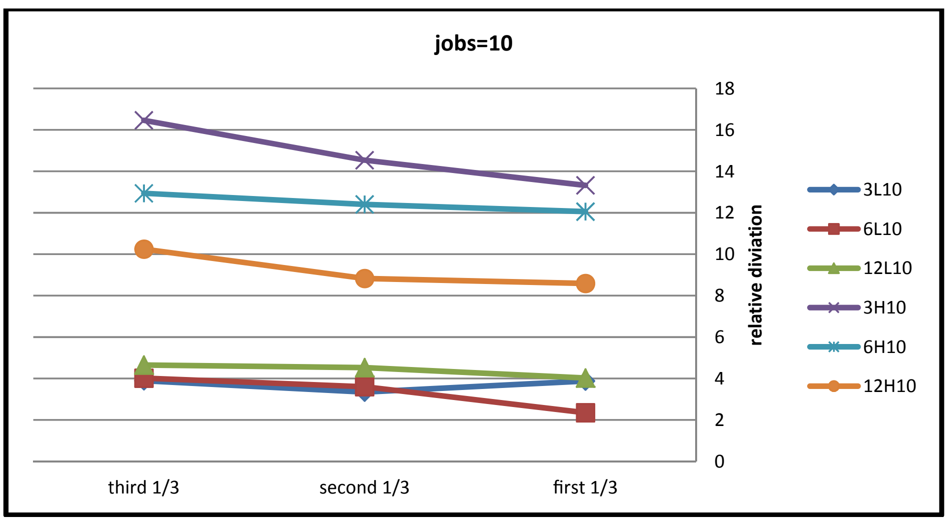

Fig. 1. Relative deviation from lower bound for jobs=10

According to Fig. 1, when the bottleneck station moves to the end of the production line, the RPD significantly increases. In addition, if the number of machine in each stage is set at level L, the RPD is lower than when it is set at level H.

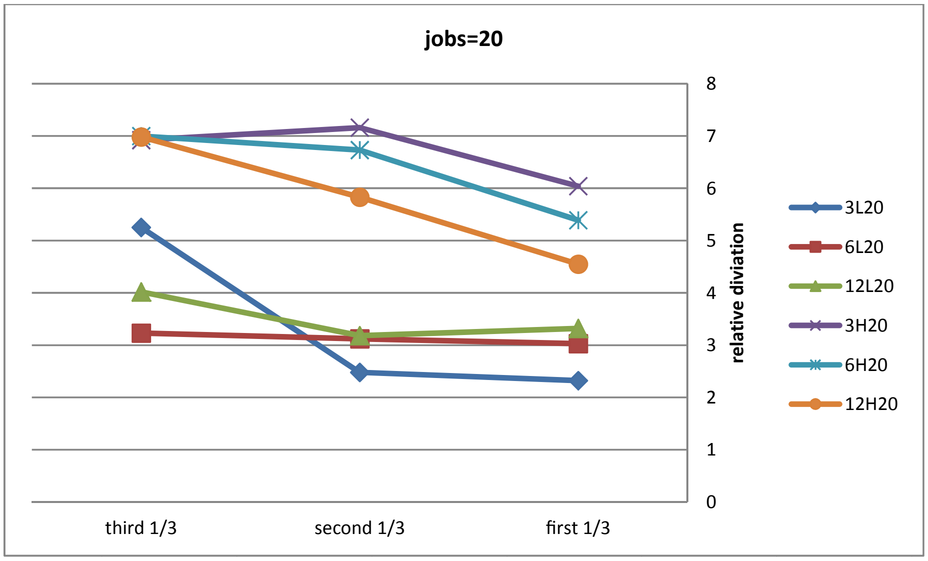

Fig. 2. Relative deviation from lower bound for jobs=20

Based on Fig. 2, if the number of machines in each stage is set at level L, the RPD is lower than level H. Also, relative deviation from the lower bound can achieve the lowest value if the bottleneck stage is located at first 1/3.

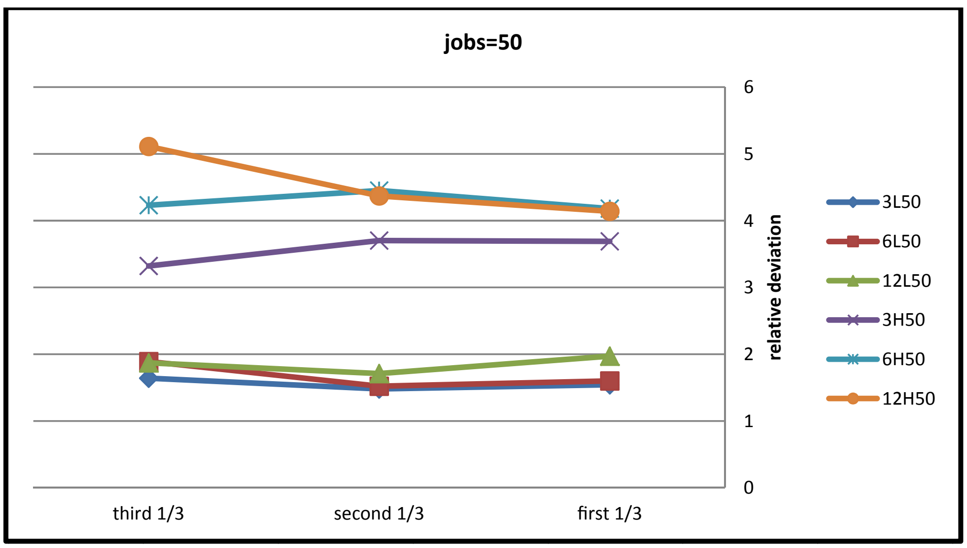

Fig. 3. Relative deviation from lower bound for jobs=50

As it can be seen in Fig. 3, if the number of machines in each stage is set at level low, the RPD is significantly lower than level H. But, we cannot see any significant pattern in the RPD when the bottleneck stage moves to each position in production line.

5-3. Comparison of the proposed heuristic with other well-known heuristics

In this section, we compare the proposed heuristic with other well-known heuristics like CDS[2], Slope and NEH[3]. These heuristics are developed for the flow shop scheduling problem. Hence, we developed these heuristics for the HFS problem with unrelated parallel machines by inserting the machine selection rule in the structure of these well-known heuristics. Table 5 presents the number of the best solutions produced by any heuristic algorithm.

Tab. 5. Number of the best solution producing by each heuristic

Algorithms	Number of jobs			Number of Stages			Number of machines in each stages		Bottleneck position			NBS
	L	M	H	L	M	H	L	H	First 1/3	Second 1/3	Third 1/3	
CDS	39	23	27	46	28	25	38	36	38	31	30	15.6%
Slope	21	25	26	26	34	21	26	16	24	26	26	11.9%
NEH	51	48	44	48	43	41	45	55	53	45	56	22.9%
Proposed Heuristic	117	129	116	123	125	115	188	199	124	117	121	62.3%

According to these results, we can rank these heuristics in terms of NBS. The proposed heuristic algorithm is the best heuristic such that it outperforms other heuristic algorithms in 62.3% of instances. Afterwards, NEH (with 22.9% of instances), CDS (with 15.6% of instances), and Slope (with 11.9% of instances) generate the best solutions.

6. Conclusion

This research conducted the proposed lower bound which can achieve the optimal solution in many instances so that the solver cannot attain the optimal solution in acceptable time in some instances, the lower bound gives a good solution. The lower bound has 2.56% difference from the lower bound in small instances. Therefore, it can be used to evaluate other heuristic performance. The heuristic algorithm is proposed in this article for the HFS scheduling problems with unrelated parallel machines, which is a good heuristic for C_{max} objective function, such that it has, on average, 5.4% deviation from the lower bound. The results show that the bottleneck-heuristic algorithm outperforms the other well-known heuristics such as CDS, Slope, and NEH.

Considering other characteristics of the HFS scheduling problems, such as sequence-dependent set up time, reentrant jobs and identical machines can be opened for future studies. In addition, new bottleneck-based approaches can be studied for other scheduling problems with bottleneck stage.

Appendix

Tab. 6. Experimental results of the test problems

Job	N.M.	N.S.	3						6						12					
		B.P.	First 1/3		Second 1/3		Third 1/3		First 1/3		Second 1/3		Third 1/3		First 1/3		Second 1/3		Third 1/3	
		T.P.	Cmax	LB	Cmax	LB	Cmax	LB	Cmax	LB	Cmax	LB	Cmax	LB	Cmax	LB	Cmax	LB	Cmax	LB
10	L	1	70	67	76	74	76	76	142	139	142	137	143	134	262	244	248	239	268	260
		2	77	70	69	69	71	68	134	133	132	123	142	135	248	238	262	251	250	244
		3	81	79	79	77	85	82	141	134	151	145	135	127	250	242	256	250	262	242
		4	78	78	75	72	68	63	149	143	145	142	144	138	280	264	262	248	284	268
		5	75	71	74	69	80	78	144	141	143	141	147	137	252	246	290	281	262	254
		6	69	66	68	64	72	71	152	147	135	130	140	131	266	260	270	253	274	260
		7	79	74	78	73	71	70	143	142	146	142	154	150	258	250	276	258	258	244
		8	75	75	82	80	80	73	146	143	151	145	144	148	258	244	260	249	254	242
		9	77	74	77	77	77	73	139	137	140	134	136	129	284	270	254	242	270	254
		10	71	71	70	69	84	82	141	139	141	137	140	134	264	260	296	284	286	282
	H	11	56	50	68	61	60	49	107	98	107	100	108	93	212	199	209	189	210	189
		12	66	58	56	52	53	47	115	104	117	110	100	85	198	175	221	207	221	204
		13	54	49	59	50	66	61	98	90	104	87	121	103	211	198	197	179	212	200
		14	60	51	63	55	60	49	103	94	103	89	96	91	222	202	203	184	199	182
		15	57	51	55	48	52	44	109	92	99	87	109	101	206	193	214	195	208	191
		16	52	45	62	50	67	60	101	91	102	92	102	89	209	189	207	193	214	201
		17	59	52	50	43	57	48	112	104	114	98	111	95	215	203	220	201	225	203
		18	63	54	59	51	64	60	104	88	119	104	118	102	201	183	211	197	220	193
		19	67	61	57	55	57	46	105	91	96	89	97	89	208	191	219	201	221	197
		20	57	51	61	53	50	42	95	84	106	94	109	100	222	204	203	188	206	179
20	L	21	138	131	139	133	151	144	289	281	310	300	322	315	399	390	408	392	399	383
		22	125	122	148	146	141	135	276	266	316	311	306	295	410	395	397	388	412	401
		23	148	145	151	150	138	130	287	280	307	299	318	311	414	403	401	390	415	395
		24	139	137	139	139	125	117	297	289	301	293	319	305	393	385	389	376	391	373
		25	134	132	135	131	134	126	281	276	299	288	301	289	399	388	411	399	412	403
		26	125	123	125	122	136	127	292	284	307	298	313	305	413	402	423	410	421	405
		27	125	122	137	132	151	144	280	266	314	301	304	296	425	404	413	395	400	380
		28	137	135	125	120	148	142	274	265	306	295	316	304	406	390	424	412	413	400
		29	145	143	148	144	139	132	276	269	301	292	311	300	394	381	407	396	395	378

		30	129	125	136	134	150	146	288	281	315	306	305	298	393	378	395	385	406	390
	H	31	108	104	111	102	110	97	225	214	222	203	211	201	431	411	421	398	484	439
		32	98	91	117	109	118	112	212	203	203	191	207	197	422	403	444	423	453	433
		33	115	107	105	98	101	96	210	196	210	201	229	205	447	422	437	413	430	405
		343	105	98	101	96	97	92	203	195	225	207	214	196	433	416	415	391	441	423
		35	108	102	99	92	109	105	213	200	213	199	215	203	422	408	448	423	447	410
		36	99	95	106	102	106	101	211	201	219	207	209	193	436	422	431	420	435	413
		37	115	110	107	97	109	99	223	206	202	191	226	212	435	416	435	407	457	422
		38	107	98	117	111	117	112	222	211	205	189	206	194	448	429	449	411	437	407
		39	101	96	107	98	111	102	215	204	212	200	211	199	434	412	431	410	441	404
		40	117	111	109	102	106	99	201	196	208	197	215	203	453	432	456	431	456	433
	L	41	320	313	328	325	309	305	668	651	698	689	690	681	1524	1491	1504	1482	1520	1495
		42	333	329	313	309	323	319	682	670	670	661	671	662	1480	1455	1486	1451	1476	1449
		43	313	310	322	318	312	308	656	643	654	646	652	639	1534	1511	1470	1443	1496	1472
		44	324	318	313	308	326	323	680	671	690	683	684	671	1510	1488	1520	1501	1528	1503
		45	337	331	326	321	317	309	692	683	676	668	676	665	1484	1453	1476	1449	1488	1455
		46	310	306	317	310	309	303	658	651	680	669	692	679	1473	1446	1532	1502	1507	1477
		47	316	313	310	306	306	301	694	687	654	641	676	665	1544	1507	1540	1521	1548	1514
		48	336	329	321	318	314	311	678	669	674	661	673	651	1492	1461	1508	1471	1499	1471
		49	329	324	330	324	331	325	656	648	667	653	696	686	1512	1481	1492	1463	1516	1487
		50	319	315	312	307	324	316	672	657	696	687	658	644	1484	1454	1468	1462	1483	1461
	H	51	224	215	225	215	213	224	478	458	466	445	451	425	1033	996	1017	975	1032	991
		52	208	201	206	201	207	208	456	443	483	470	483	470	1012	963	1013	972	1016	962
		53	218	211	213	208	231	218	468	452	477	449	472	462	998	964	1022	980	1025	970
		54	227	219	217	212	217	227	442	421	455	433	498	472	1037	990	1006	961	1005	953
		55	210	202	229	219	203	210	470	453	472	448	460	446	1010	962	991	950	1011	965
		56	203	196	216	206	225	203	491	476	463	445	456	435	992	955	1007	973	995	948
		57	218	211	207	200	219	218	473	449	482	461	473	453	1002	961	995	956	1016	969
		58	229	221	218	209	229	229	452	435	473	452	484	469	1014	978	1031	991	1034	988
		59	224	211	211	203	213	224	466	445	491	469	467	449	1028	982	1016	964	1016	964
		60	219	215	219	211	221	219	488	464	463	452	455	429	993	966	1001	954	1025	971

N.M.: number of machines in each stage; N.S.: number of stages; B.P.: bottleneck position; T.P: test problem; Cmax: $C_{heuristic}$; LB: C_{LB}

References

[1] Kurz, M.E. and Askin, R.G. "An adaptable problem-space-based search method for flexible flow line scheduling", IIE Transaction, Vol. 33, No. 8, (2001), pp. 691–693.

[2] Ramezanian, R., Aryanezhad, M.B. and Heydari, M. “A Mathematical Programming Model for Flow Shop Scheduling Problems for Considering Just in Time Production”, International Journal of Industrial Engineering & Production Research, Vol. 21, No. 2, (2010), pp. 97-104.

[3] Ribas, I., Leisten, R., Framinan, J.M. “Review and Classification of Hybrid Flow Shop Scheduling Problems from a Production System and a Solutions Procedure Perspective” Computers and Operation Research, Vol. 37, (2009), pp.1439–1454.

[4] Lin, J.T., and Chen, C.M. “Simulation optimization approach for hybrid flow shop scheduling problem in semiconductor back-end manufacturing” Simulation Model Practice and Theory, Vol. 51, (2015), pp. 100-114.

[5] Agnetis, A., Pacifici, A., Rossi, F., Lucertini, M., Nicoletti, S., Nicolo, F., “Scheduling of Flexible Flow Shop in an Automobile Assembly Plant” European Journal of Operation Research, Vol. 97, No. 2, (1997), pp. 348–362.

[6] Adler, L., Fraiman, N., Kobacker, E., Pinedo, M., Plotnicoff, J.C. and Wu, T.P. “BPSS: a Scheduling Support System for the Packaging Industry”, Operations Research, Vol. 41, (1993), pp.641–648.

[7] Voss, S., Witt, A. “Hybrid Flow Shop Scheduling as a Multi-Mod Multi-Project Scheduling Problem with Batching Requirements: a Real-World Application” International Journal of Production Research, Vol. 105, No. 2, (2007), pp. 445–458.

[8] Jin, Z.H., Ohno, K., Ito, T., Elmaghraby, S.E., “Scheduling Hybrid Flowshops in Printed Circuit Board Assembly Lines” Production and Operation Management, Vol. 11, No. 2, (2002), pp. 216–230.

[9] Jenabi, M., Fatemi Ghomi, and S.M.T., Torabi, S.A., and Karimi B. “Two hybrid meta-heuristics for the finite horizon ELSP in flexible flow lines with unrelated parallel machines” Applied Mathematics and Computation, Vol. 186, (2007), 230–245.

[10] Zabihzadeh, S. and Rezaeian, J. “Two meta-heuristic algorithms for flexible flow shop scheduling problem with robotic transportation and release time”, Applied Soft Computing, Vol. 40, (2016), pp. 319-330.

[11] Sawik, T., “An Exact Approach for Batch Scheduling in Flexible Flow Lines with Limited Intermediate Buffers” Mathematical Computation Model. Vol. 36, Nos. 4–5, (2002), pp. 461–471.

[12] Sahraeian, R. “A Two-Stage Hybrid Flowshop Scheduling Problem with Serial Batching”, International Journal of Industrial Engineering & Production Research, Vol. 25, No. 1, (2014), pp. 55-63.

[13] Rabiee, M.m Jolaim F., Asefi, H,, Fattahi, P. and Lim, S. “A biogeography-based optimization algorithm for a realistic no-wait hybrid flow shop with unrelated parallel machines to minimise mean tardiness”, International journal of Computer Integrated Manufacturing, Vol. 29, No. 9, (2016), pp. 1007-1024.

[14] Botta-Genoulaz, V., “Hybrid Flow Shop Scheduling with Precedence Constraints and Time Lags to Minimize Maximum Lateness” International Journal of Production Economic, Vol. 64, Nos. 1–3, (2000), pp. 101–111.

[15] Cheng, J., Yoshiyuki, K., Kise, H., “A Shifting Bottleneck Approach for Parallel-Machine Flowshop Scheduling Problem”, Journal of the Operation Research Society of Japan, Vol. 44, No. 2, (2001), pp. 141–154.

[16] Quadt, D., Kuhn, H., “A Taxonomy of Flexible Flow Line Scheduling Procedures” European Journal of Operation Research, Vol. 178, No. 3, (2007), pp. 686–698.

[17] Nahavandi, N. and Asadi Gangraj, E. "A New lower bound for Flexible Flow Shop Problem with Unrelated Parallel Machines", International Journal of Industrial Engineering & Production Research, Vol. 25, No. 1, (2014), pp. 65-70.

[18] Torabi, S.A., Ghomi, S.M.T.F., Karimi, B., "A Hybrid Genetic Algorithm for the Finite Horizon Economic Lot and Delivery Scheduling in Supply Chains, European Journal of Operation Research, vol. 173, No. 1, (2006), pp. 173–189.

[19] Vignier, A., Billaut, J.C., & Proust, C. "Hybrid flowshop scheduling problems: state of the art" Rairo-Recherche Operationnelle-Operations Research. Vol. 33, No. 2, (1999), pp.117–183.

[20] Kis, T., and Pesch, E. "A review of exact solution methods for the non-preemptive multiprocessor flowshop problem" European Journal of Operation Research, Vol. 164, No. 3, (2005), pp. 592–608.

[21] Goldratt, E., and Cox J. "The goal: a process of ongoing improvement" New York: North River Press, 1992.

[22] Goldratt, E., and Fox, R. (2004). The race. New York: North River Press.

[23] Acero-Dominguez, M., Paternina-Arboleda, C. "Scheduling jobs on a k-stage flexible flow shop using a TOC-based (bottleneck) procedure" Proceedings of the 2004 Systems and Information Engineering Design Symposium, (2004), pp:295-298.

[24] Chen, C.L., and Chen, C.L. "A bottleneck-based heuristic for minimizing makespan in a flexible flow line with unrelated parallel machines, Computers & Operations Research, Vol. 36, (2009), pp. 3073-3081.

[25] Paternina-Arboleda, C., Montoya-Torres, J., Acero-Dominguez, M. and Herrera-Hernandez, M. "Scheduling jobs on a k - stage flexible flow-shop" Annals Operation Research, Vol. 164, No. 1, (2007), pp. 29–40.

[26] Pinedo, M. "Scheduling: Theory, algorithms, and systems" Englewood Cliffs: Prentice-Hall, (2007).

[27] Chen, C.L., and Chen, C.L. "Bottleneck-based heuristics to minimize total tardiness for the flexible flow line with unrelated parallel machines" Computers & Operations Research, Vol. 56, (2009), pp. 1393-1401.

[28] Haouari, M., M'Hallah, R., "Heuristic Algorithm for Two Stage Hybrid Flow Shop", Operation Research Letter, Vol. 21, (1997), pp. 43-53.

[29] Santos, D.L., Hunsucker, J.L., and Deal, D.E. "Global lower bounds for flow shops with multiple processors" European journal of operational research, Vol. 80, (1995), pp. 112-120.

[30] Soewandy, H., Elmaghrby, S.E., "Sequencing Three- Stage Flexible Flowshops with Identical Machines to Minimize Makespan", IIE transactions, Vol. 33, (2001), pp. 985-993.

[31] Leon, V.J. and Ramamoorthy, B. "An adaptable problem-space-based search method for flexible flow line scheduling", IIE Transactions, Vol. 29, (1997), pp. 115-125.

[32] Tadayon, B., and Salmasi, N. "A two-criteria objective function flexible flowshop scheduling problem with machine eligibility constraint" International Journal of Advanced Manufacturing Technology, Vol. 64, (2013), pp. 1001–1015.

Follow This Article at The Following Site

Asadi Gangraj E. Heuristic approach to solve hybrid flow shop scheduling problem with unrelated parallel machines. IJIEPR. 2017; 28 (1) :61-74
URL: http://ijiepr.iust.ac.ir/article-1-718-en.html
DOI: 10.22068/ijiepr.28.1.61