IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 29 March 2023, accepted 15 April 2023, date of publication 18 April 2023, date of current version 2 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3268214

RESEARCH ARTICLE

An Evaluation of Mathematical Programming and Lower-Bound Methods for Hybrid Flow Shop Problems With a Makespan Criterion

YARONG CHEN[1], YA-CHIH TSAI[2], AND FUH-DER CHOU[1]
[1]College of Mechanical and Electrical Engineering, Wenzhou University, Wenzhou, Zhejiang 325035, China
[2]Department of Hotel Management, Vanung University, Taoyuan City 32061, Taiwan

Corresponding author: Fuh-Der Chou (fdchou@tpts7.seed.net.tw)

This work was supported by the National Natural Science Foundation of China under Grant 51705370.

ABSTRACT This paper considers the hybrid flow shop scheduling problem, where jobs are processed in m stages with the same route of the stage. Each stage has identical parallel machines for processing jobs. Some mathematical programming formulations and lower bound calculations have been proposed in the literature for such cases. Nevertheless, there is a lack of complete comparisons of these mathematical programming formulations and lower bounds in the hybrid flow shop literature. This paper proposes a new mixed integer programming model and two new lower bounds based on the bin-packing concept for the considered problem. To evaluate the proposed model, two sets of small and small-to-medium problems are used to compare our model with the existing models. Moreover, two propositions are discussed for lower bounds. The experimental results show that the proposed mixed integer programming model efficiently found optimal solutions because it needs a smaller number of binary variables and a smaller number of constraints, and the proposed lower bound can also serve as a strong indicator to evaluate the distances between the solutions obtained by heuristic algorithms and the optimal solution.

INDEX TERMS Hybrid flow shop, makespan, mixed integer programming, lower bound.

I. INTRODUCTION

The pioneering study on flow shop scheduling problems is that of Johnson [1]. This type of flow shop problem (FSP) has received increasing attention from researchers. In the FSP, a set of jobs flow through multiple stages in the same order, each stage having only one machine. In the literature, various FSP extensions addressing industry-specific situations have been developed. For example, the distributed permutation flow shop scheduling problem (DPFSP) with multi-factory manufacturing is an extension of the FSP in response to the development of globally distributed production [1]. The hybrid FSP (HFSP) is another well-known example of FSP generalization motivated by the fact that parallel machines are usually required in the flow stages to prevent the production system from being blocked by the unavailability (e.g., breakdown) of a single machine. Additionally, multiple identical machines are added at some given stages in a way that not only increases the overall throughput of the shop but also further reduces the impact of the bottleneck stage on the overall shop efficiency [2]. Numerous applications of HFSPs have been studied in the literature. These include industries as diverse as textile processing [3], glass and paper making [4], furniture manufacturing [5], plastic manufacturing [6], and steel making [7], [8].

In recent years, more complex variant FSPs combining multiple factories and parallel machines have attracted many researchers [1], [9], [10]. Overall, the relationship of variants of FSPs can be illustrated in Fig. 1.

In this paper, we consider HFSPs. Recall that, these problems consist of n jobs that are processed in a flow shop, following the same route of the stages; i.e., the jobs are first

The associate editor coordinating the review of this manuscript and approving it for publication was Shih-Wei Lin.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

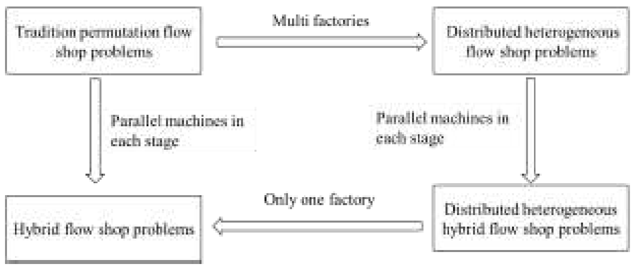

FIGURE 1. The relationship among variants of FSPs.

processed in stage 1, then stage 2, and so on until the last stage, and more than one machine is necessary for at least one stage.

Our objective is to minimize the makespan. The makespan is related to the maximization of machine utilization or system throughput [11] and has been studied very intensively in the literature. Employing the three-field notation $\alpha/\beta/\gamma$ [12], the deterministic HFSP minimizing the makespan can be defined as $HF_k//C_{max}$, where HF_k specifies a k-stage hybrid flow shop production system and C_{max} refers to the makespan. Since effective resource allocation and task sequences play a key role in manufacturing systems to achieve the company's goal, good scheduling is very important.

Mathematical programming modeling in the form of mixed integer programming (MIP) is an essential tool for understanding the problem characteristics and obtaining optimal solutions by formulating constraints and objective functions explicitly. In our search for MIP models solving $HF_k//C_{max}$ problems in the literature, all except one of the studies we found only formulate MIP model, without giving a comparison with other models. The only exception was the study proposed by Naderi et al. [13], in which the performance of four different MIP models was evaluated. However, in their comparison, other MIP models for $HF_k//C_{max}$ problems were not included.

In the scheduling literature, in addition to the optimal solutions, lower bounds are frequently used as benchmarks for evaluating the performance of heuristic or metaheuristic algorithms. For $HF_k//C_{max}$ problems, three lower-bound calculations based on different problem relaxations, which will be described later he Section III, have been proposed in the literature [14], [15]. Although the global lower bound proposed by Santos et al. [14] has often been adopted in many HFSP studies, there is still room to develop other powerful lower bounds.

Our study considers the problem of $HF_k//C_{max}$ due to its significance in both NP-hard features and production applications, and we formulate a new MIP model and two lower-bound methods. In addition, we adopt the same testbeds proposed by Fernandez-Viagas et al. [16] to fairly examine the performance of the MIP models and obtain a detailed benchmark for all instances in the testbeds. Regarding lower bounds, we make two propositions to analyze their dominances and further suggest a better lower bound calculation.

The rest of this paper is organized into five sections. In Section II, we review previous related HFSPs. We define the considered problem and assumptions in Section III. Different existing MIP models are also preliminarily examined to show their verification and validation. Moreover, our MIP model is proposed here. In Section IV, we discuss three different lower bounds and then propose two new lower-bound methods based on the bin-packing concept. The dominance properties associated with these lower bounds are also proposed. In Section V, numerical experiments are carried out, and the results are presented. Finally, some concluding remarks are given in Section VI.

II. LITERATURE REVIEW

Starting with the study of Arthanari and Ramamurthy [17], the area of HFSP scheduling problems has grown considerably, and currently, the variants of HFSP scheduling problems in the literature are vast. First, we focus on our considered problem, i.e., $HF_k//C_{max}$. The $HF_k//C_{max}$ problem is theoretically NP-hard, as shown by Gupta, even when the problem has only two stages, and one of the stages contains a single machine [18]. Some existing solution approaches, such as branch-and-bound methods [17] and heuristic algorithms [18], [19], [20], [21], [22], [23], have been developed for the $HF_k//C_{max}$ problem with two or three stages [24].

Regarding HFSPs with more than three stages, Brah and Hunsucker [25] developed the branch-and-bound (*B&B*) method to find optimal solutions, and later, some improved *B&B* methods were designed to optimally solve a larger range of HFSPs [26], [27], [28]. In addition to *B&B* methods, MIP models have been utilized to solve HFSPs by many researchers [29], [30], [31]. Since these problems are NP-hard, the *B&B* methods and MIP models can only solve small-sized problems. Consequently, some well-known heuristic algorithms used to solve traditional permutation flow shop problems have been modified to solve HFSPs [25], [32], [33]. Regarding metaheuristic algorithms, Nowicki and Smutnicki [34] developed a tabu search (TS) algorithm, which was one of the first metaheuristic to solve large-scale HFSPs. Alaykyran et al. [35] proposed an ant colony optimization (ACO) method and showed that this algorithm is quite competitive compared to the *B&B* method of Néron et al. [28]. Engin and Döyen [36] proposed an artificial immune system (AIS) algorithm for solving HFSPs to minimize the makespan. Liao et al. [4] presented particle swarm optimization (PSO) combined with a bottleneck heuristic. Later, the discrete artificial bee colony (DABC), proposed by Pan et al. [37], outperformed the PSO of Liao et al. [4] and the AIS of Engin and Döyen [36]. Marichelvam et al. [5] solved HFSPs using the cuckoo search algorithm. Kahraman et al. [38] developed a genetic algorithm (GA) and compared it with the AIS and *B&B* of Carlier and Neron [27], with 1600 s as a termination criterion. Kizilay et al. [39] proposed an iterated greedy (IG) method and showed that DABC was the best-performing method and that IG outperformed AIS, with $50 \cdot n \cdot m$ milliseconds as a stopping condition. Many metaheuristic algorithms require algorithm-specific parameters to be tuned to attain high-quality solutions, which is rather sensitive and time-consuming. Hence, Buddala and Mahapatra [40] proposed the teaching-learning-based optimization (TLBO) and JAYA algorithms, which do not have any algorithm-specific tuning parameters, to solve the HFSP.

Some researchers have also considered other factors of HFSPs; for example, Jiang et al. considered a three-stage HFSP motivated by steelmaking and continuous casting manufacturing, in which processing continuity, setup time, and intraflow constraints are involved [41]. Qi et al. considered job family and sequence-dependent setup times for the HFSP and developed a MIP and an IG algorithm [42]. Liu et al. considered multiskilled workers and fatigue factors for the HFSPs and developed simulation-based optimization (SBO) to solve these problems [43]. Other HFSPs involving different constraints can be found in [44].

Compared with HFSPs, where a single factory is considered, research on distributed HFSP scheduling has increased recently since the multifactory manufacturing strategy to enhance a company's competitiveness was introduced to production systems [1], [9], [10], [45], [46].

As stated earlier, we focus on the $HF_k//C_{max}$ problem. For the considered problem, different MIP models and lower bounds are proposed in the literature; however, they are likely to be compared using different conditions or not compared with other methods. Motivated by the study of Naderi et al. [13] and the testbeds proposed by Fernandez-Viagas et al. [16], this study develops a MIP model for $HF_k//C_{max}$ problems to obtain optimal solutions as benchmarks. It provides a fair comparison with the existing MIP models for the $HF_k//C_{max}$ problem using the same test instances. Moreover, we propose two lower-bound methods as strong indicators when the optimal solutions cannot be obtained.

III. PROBLEM DEFINITION AND RELEVANT MATHEMATICAL PROGRAMMING MODELS

The HFSP can be defined as follows. The set of n jobs is to be processed in each stage in order from stage 1 to stage k, following the same route. In stage k, there are M_k identical machines, each machine can only process one job at a time, and each job can be processed by at most one machine at a time. The processing time of job j when it is processed in stage k is denoted by p_{jk}. Other assumptions that are usually used for the HFSP under consideration are as follows [31]:

- No preemption is allowed.
- All jobs are ready for processing at the same time (i.e., the release times of the jobs are set to 0).
- All machines are available in the whole scheduling planning horizon.
- Transportation times are either insignificant or constant.
- Setup times are considered insignificant.
- The inventory buffer between stages is unlimited.

TABLE 1. Processing times for an example problem.

Job	Stage 1	Stage 2	Stage 3	Stage 4
1	66	53	20	87
2	37	81	40	92
3	54	48	37	97
4	52	81	23	43

TABLE 2. Results of examining ten existing models using the CPLEX solver.

Model	Reference paper	Execution? (Verification)	Correct? (Validation)
MIP*(KP)	Kis and Pesch [29]	Yes	No
MIP(K)	Kahraman et al. [38]	Yes	Yes
MIP(P)	Paternina-Arboleda et al. [31]	No	No
MIP(L)	Liao et al. [4]	Yes	No
MIP*(PD)	Pan and Dong [47]	Yes	No
MIP(N1)	Naderi et al. [13]	Yes	Yes
MIP(N2)	Naderi et al. [13]	Yes	Yes
MIP(N3)	Naderi et al. [13]	Yes	Yes
MIP*(N4)	Naderi et al. [13] and Fernandez-Viagas et al. [16]	Yes	No
MIP(LC)	Lin et al. [30]	No	No

We first review ten models suggested by researchers between 2005 and 2021 and determine whether they can execute and obtain correct optimal solutions with the CPLEX solver. Based on the results, it is found that the two models proposed by Paternina-Arboleda et al. [31] and Lin et al. [30] are incomplete; for example, constraints (5) and (6) in Lin's model [30] lack an extremely large number that would force these constraints to be reasonable. In addition, some models cannot guarantee that they will obtain optimal solutions even though they may work. For example, the model of Kis and Pesch [29] solves an instance with four jobs afourd four stages, where all stages have three machines; as shown in Table 1, the obtained solution is 213, which is not equal to the optimal solution, 269. Table 2 shows the results after executing the models using the CPLEX solver. The first column denotes the corresponding model from the reference paper, where the symbol "∗" indicates that the model has been modified in this paper to obtain optimal solutions. For details about the MIP models, we refer the interested reader to the corresponding studies.

A. MIXED INTEGER LINEAR PROGRAMMING MODEL

Before mathematically modeling the HFSP, the indices, parameters, and decision variables are defined, as shown in Table 3.

The mixed integer programming model adapted from Naderi et al. [13] can be stated as follows:

$$\text{Minimize } C_{max} \quad (1)$$

$$\text{Subject to } Y_{jhk} + Y_{hjk} \leq 1, j, h = 1, \ldots, n;$$

$$j \neq h or j < h; k = 1, \ldots, m \quad (2)$$

$$S_{jm} + p_{jm} \leq C_{max}, j = 1, \ldots, n \quad (3)$$

TABLE 3. Indices, parameters, and decision variables.

Symbol	Description
i	The index of a machine.
j, h	Indices of jobs.
n	The number of jobs.
m	The number of stages.
M_k	The number of machines in stage k, $k = 1,2,\dots,m$.
p_{jk}	The processing time of job j in stage k.
Big$M$	An extremely large positive integer, where Big$M$ defaults to 32767.
S_{jk}	The start time of job j for processing in stage k, $j = 1,2,\dots,n$; $k = 1,2,\dots,m$.
X_{jki}	A binary variable, 1 if job j is processed by machine i on stage k; 0 otherwise. $j = 1,2,\dots,n$; $k = 1,2,\dots,m$; $i = 1,2,\dots,M_k$.
Y_{jhk}	A binary variable, 1 if job h is processed immediately after job j by machine i in stage k; 0 otherwise. $j,h = 1,2,\dots,n$; $k = 1,2,\dots,m$; $j \neq h$
C_{max}	The maximum completion time of all jobs.

$$\sum\nolimits_{i=1}^{M} {}_k X_{jki} = 1, \quad j = 1,\dots,n;\ k = 1,\dots,m \tag{4}$$

$$S_{jk} + p_{jk} \le S_{j,k+1}, \quad j = 1,\dots,n; \quad k = 1,\dots,m-1 \tag{5}$$

$$(Y_{jhk} + Y_{hjk} - 1) + BigM \times (2 - X_{jki} - X_{hki}) \ge 0 \quad j,h = 1,\dots,n, j < h; \quad k = 1,\dots,m; \quad i = 1,\dots,M_k \tag{6}$$

$$S_{hk} - (S_{jk} + p_{jk}) + BigM \times (1 - Y_{jhk}) \ge 0 \quad j,h = 1,\dots,n; j < h; \quad k = 1,\dots,m \tag{7}$$

$$\sum\nolimits_{k=1}^{K} \sum\nolimits_{i=1}^{M} {}_k X_{jki} = n \quad k = 1,\dots,m \tag{8}$$

$$S_{jk} \ge 0 \tag{9}$$

$$X_{jki} \in \{0, 1\} \tag{10}$$

$$Y_{jkh} \in \{0, 1\} \tag{11}$$

In this model, $BigM$ is a very large constant, i.e., greater than the sum of all job processing times. The makespan minimization of the considered problem is expressed by (1). Inequalities (2) imply that at each stage, there is a pair of jobs (j, h) such that job j is only processed before job h or after job h. Inequalities (3) ensure that the completion times of all jobs at the last stage are less than or equal to the objective makespan. Constraints (4) ensure that each job is processed on a machine once at each stage. Constraints (5) ensure that the process of each job in one stage starts after the process completed in the previous stage. Constraints (6) state the relation between any pair of jobs on the same machine at each stage with respect to the sequence, i.e., that one is a predecessor of the other or vice versa. Constraints (7) state the relation between any pair of jobs on the same machine at each stage that forces the machine to process one job at a time. Constraints (8) ensure that each job goes through the stages. Finally, constraints (9), (10), and (11) define the decision variables.

B. THE COMPLEXITY OF THE MODELS

Suppose that there are n jobs, m stages, and M_k machines in each stage k. The proposed model is compared with the above models, excluding MIP(P), using a size complexity evaluation proposed by Naderi [13]. The size complexity evaluation includes the numbers of binary variables, continuous variables, and constraints. The comparison results are shown in Table 4. From Table 4, it can be found that MIP(K), MIP*(N4), and MIP(C) (the proposed model) have few binary variables, integer variables, and constraints. Due to this fact, the computational burdens of these three models will be lower, and they will be more efficient, which is shown in the experiment section later.

IV. LOWER BOUNDS

A. SIMPLE LOWER BOUND

An intuitive job-based lower bound is based on the idea that no job can finish processing earlier than its total processing time, i.e., $\sum_{k=1}^{m} p_{jk}$. Thus, the simple lower bound is: $LB_0 = \max_{1\le j\le n} \{\sum_{k=1}^{m} p_{jk}\}$, and it can be computed in $O(n)$ time [14].

The second lower bound is modified from the lower bound derived for the scheduling problem of identical parallel machines with heads and tails [14]. The lower bound of Carlier [15] is defined as $LB_c(J) = \min_{j\in j} r_j + \left\lceil \frac{1}{m} \sum_{j\in J} p_j \right\rceil \min_{j\in j} + q_j$. Inspired by Carlier's lower bound, we generate a set of m artificial problems for identical parallel machines with heads and tails from the original m-stage HFSP. Next, the bound is obtained using the lower bound of Carlier [15] for each of the m subproblems; the maximum bound among the m subproblems becomes the lower bound of the considered problem. Thus, the second lower bound is:

$$LB_1 = \max_{1\le k\le m} \{LB - 1(k)\}$$

where $LB_1(k) = H_1(k) + B_1(k) + T_1(k)$. $H_1(k)$, $B_1(k)$, and $T_1(k)$ indicate the heads, bodies (identical parallel machines), and tails for stage k, $k = 1,\dots,m$, respectively, and are described as follows:

$$H_1(k) = \begin{cases} 0 & k = 1 \\ \min_{1\le j\le n} \left\{\sum_{i=1}^{k-1} p_{ji}\right\} & k > 1; \end{cases}$$

$$B_1(k) = \left\lceil \frac{1}{M_k} \left(\sum\nolimits_{j=1}^{n} p_{jk}\right) \right\rceil;$$

$$T_1(k) = \begin{cases} \min_{1\le j\le n} \left\{\sum_{i=k+1}^{m} p_{ji}\right\} & k < m \\ 0 & k = m. \end{cases}$$

B. A STAGE-BASED LOWER BOUND

The stage-based lower bound procedure was developed by Santos et al. [14] and has been adopted by many researchers to evaluate their proposed algorithms in the literature [4], [25], [31], [37], and [40]. The procedure employed for the stage-based lower bound is similar to that for the second lower bound mentioned above. However, the stage-based lower bound is based on the concept of averaging; that is, for each stage k, the average bound of the jobs processed

TABLE 4. **Comparing different models using a size complexity evaluation (Naderi [13]).**

Model	Number of binary variables	Number of continuous variables	Number of constraints
MIP*(KP)	$n(n+1)m\left(\sum M_k\right)$	nm	$nm[(n-1)+2\sum M_k]+\sum M_k-n$
MIP(K)	$n(n+1)m\left(\sum M_k\right)$	nm	$nm\left[2+3(n-1)\sum M_k\right]$
MIP(L)	$nm\left(n-1+\sum M_k\right)$	$2nm$	$nm(n+3)-n$
MIP*(PD)	$nm\left(n-1+\sum M_k\right)$	nm	$nm\left[(n+1)+(n-1)\sum M_k\right]$
MIP(N1)	$n^2m\left(\sum M_k\right)$	nm	$nm\left[3+2n+\sum M_k\right]+n\sum M_k+2n$
MIP(N2)	$n(n-1)m\left(\sum M_k\right)+nm\left(\sum M_k\right)$	$2nm\left(\sum M_k\right)$	$nm\left[2+(2n+1)\sum M_k\right]$
MIP(N3)	$nm\left(n+\sum M_k\right)$	$2nm$	$nm\left[4+2n+\frac{1}{2}(n-1)\sum M_k\right]$
MIP*(N4)	$nm\left(n+\sum M_k\right)$	nm	$nm\left[2+(n-1)\sum M_k\right]$
MIP(LC)	$nm\left(n-1+\sum M_k\right)$	nm	$nm\left[1+(2n-1)\sum M_k\right]$
MIP(C)	$nm\left(n-1+\sum M_k\right)$	nm	$nm\left[2+(n-1)\sum M_k\right]+m$

on M_k parallel machines must be less than or equal to the maximum bound. Thus, the third lower bound is:

$LB_2 = \max_{1\le k\le m} \{LB_2(k)\}$, where

$$LB_2(k) = \left\lceil \frac{1}{M_k}(H_2(k) + B_2(k) + T_2(k)) \right\rceil;$$

$$H_2(k) = \begin{cases} 0 & k = 1 \\ \sum_{y=1}^{M_k} LSA_y(k) & k > 1 \end{cases}$$

where $LSA_y(k)$ is sequenced in increasing order of $LS_j(k)$:

$$LS_j(k) = \sum_{i=1}^{k-1} p_{ji}.$$

$$B_2(k) = \sum_{j=1}^{n} p_{jk};$$

$$T_2(k) = \begin{cases} \sum_{y=1}^{M_k} RSA_y(k) & k < m \\ 0 & k = m, \end{cases}$$

where $RSA_y(k)$ is sequenced in increasing order of $RS_j(k)$:

$$RS_j(k) = \sum_{i=k+1}^{m} p_{ji}.$$

Since $\frac{1}{M_k}H_2(k)$ and $\frac{1}{M_k}T_2(k)$ are obviously greater than or equal to $H_1(k)$ and $T_1(k)$, LB_2 dominates LB_1. For more details concerning the theorem and proof, please refer to Santos et al. [14].

C. A BIN-PACKING-BASED LOWER BOUND

Considering the lower bound of Carlier [15] and the stage-based lower bound procedure of Santos et al. [14], we could relax the considered problem to k parallel machine subproblems with heads and tails and derive a lower bound procedure based on these subproblems. That is,

$$LB_3 = \max_{1\le k\le m} \{LB_2(k)\},$$

where

$$LB_3(k) = H_3(k) + B_3(k) + T_3(k).$$

The middle part of $B_3(k)$ for each stage is regarded as an identical parallel machine problem with a makespan, i.e., $P//C_{max}$. The $P//C_{max}$ problem is closely related to the bin-packing problem, and Dell'Amico and Martello [48] proposed a better lower bound based on the bin-packing concept. Thus, we modify the lower bound of Dell'Amico and Martello [48] for $B_3(k)$, and the procedure for $B_3(k)$ is:

Step 1. Let $BLB_0(k) = \frac{1}{M_k}\sum_{j=1}^{n} p_{jk}$ and $BLB_1(k) = \max_{1\le j\le n} p_{jk}$ for $j = 1,\ldots,n$

Step 2. Sort the jobs in descending order of p_{jk}, and use $\overline{p_y}$ to denote the processing time of the job in position y of the order.

Step 3. Set the value of $BLB_2(k)$ based on the following:

$$BLB_2(k) = \begin{cases} \overline{p_1} & M_k > (n-1) \\ \overline{p_{M_k}} + \overline{p_{M_k+1}} & otherwise \end{cases}$$

Step 4. Let $L = max(BLB_0(k), BLB_1(k), BLB_2(k))$.

Step 5. Set the value of $\overline{p}$ based on the following:

$$\overline{p} = \begin{cases} \overline{p_n} & M_k > (n-2) \\ \overline{p_{M_k+2}} & otherwise \end{cases}$$

Step 6. Set the three sets of jobs based on the following:

$$J_A = \{y \mid L - \overline{p} < \overline{p_y}\}$$

$$J_B = \left\{y \mid \frac{L}{2} < \overline{p_y} \le L - \overline{p}\right\}$$

$$J_C = \left\{y \mid \overline{p} \le \overline{p_y} \le \frac{L}{2}\right\}$$

Step 7. Calculate the values of $B_\alpha(L,\overline{p})$ and $B_\beta(L,\overline{p})$:

$$B_\alpha(L,\overline{p}) = |J_A| + |J_B| + max(0, \left\lceil \frac{\sum_{y\in J_c}\overline{p_y} - \left(L\times|J_B| - \sum_{y\in J_B}\overline{p_y}\right)}{L} \right\rceil)$$

$$B_\beta(L,\overline{p}) = |J_A| + |J_B|$$

$$+ max(0, \left\lceil \frac{|J_c| - \sum_{y \in J_B} \left\lfloor \frac{L-p_y}{\overline{p}} \right\rfloor}{\left\lfloor \frac{L}{\overline{p}} \right\rfloor} \right\rceil)$$

Step 8. If $B_\alpha (L, \overline{p}) > M_k$ or $B_\beta (L, \overline{p}) > M_k$, set $L = L + 1$ and go to Step 6.

Step 9. Set $B_3(k) = L$ and stop the procedure.

The first and last parts, i.e., $H_3(k)$ and $T_3(k)$, are the same as $H_1(k)$ and $T_1(k)$ mentioned above. That is,

$$H_3(k) = H_1(k) = \begin{cases} 0 & k = 1 \\ \min_{1 \le j \le n} \left(\sum_{i=1}^{k-1} p_{ji} \right) & k > 1. \end{cases}$$

$$T_3(k) = T_1(k) = \begin{cases} \min_{1 \le j \le n} \left(\sum_{i=k+1}^{m} p_{ji} \right) & k < m \\ 0 & k = m. \end{cases}$$

D. THE MODIFIED BIN-PACKING-BASED LOWER BOUND

The fifth lower bound is modified from the bin-packing-based lower bound mentioned above.

$$LB_4 = \max_{1 \le k \le m} \{LB_4(k)\}$$

The first part, $H_{4\_E}(k)$, implies that each job should be processed through stage k-1, and there is no easy way to determine the ready time of each machine for processing jobs at the current stage k; however, we know that the ready time of each machine at the current stage k should be greater than or equal to $\min_{1 \le j \le n} \{\sum_{i=1}^{k-1} p_{ji}\}$. That is,

$$H_{4\_E}(k) = \begin{cases} 0 & k = 1 \\ \min_{1 \le j \le n} \{\sum_{i=1}^{k-1} p_{ji}\} & k > 1 \end{cases}$$

Similarly, the earliest completion time at the current stage k results in $B_{4\_E}(k) = \min_{1 \le j \le n} \{\sum_{i=1}^{k} p_{ji}\}$.

Now, consider the estimated latest finish time at the current stage k. Each stage k without a head and tail can be regarded as a $P//C_{max}$ problem, and the lower bound can be obtained by $B_3(k)$ mentioned above. Combining $H_{4\_E}(k)$ with $B_3(k)$, we can obtain the first estimated latest finish time at the current stage k. The second estimated latest finish time is adopted from LB_0, which is intuitive; that is, at the current stage k, the entire set of jobs cannot finish earlier than the total processing time for the longest-duration job, i.e., $\{\sum_{i=1}^{k} p_{ji}\}$. Consequently, we have the following as an estimated latest finish time at the current stage k:

$$B_{4\_L}(k) = \max \left(H_{4\_E}(k) + B_3(k), \max_{1 \le j \le n} \left\{ \sum_{i=1}^{k} p_{ji} \right\} \right)$$

Regarding the last part, $T_4(k)$, we use $T_{4\_max}(k)$ and $T_{4\_min}(k)$ to indicate the maximum and minimum total processing times needed from stage k to the last stage m as follows:

$$T_{4\_max}(k) = \begin{cases} \max_{1 \le j \le n} \left\{ \sum_{i=k+1}^{m} p_{ji} \right\} & k < m \\ 0 & k = m. \end{cases}$$

TABLE 5. Lower bound values.

	LB_0	LB_1	LB_2	LB_3	LB_4
Lower bound value	250	246	251	269	269

TABLE 6. The parameter set of each benchmark problem.

Instance sets	The number of jobs	The number of stages	The number of machines in each stage
Small problem (α_1)	3	$m \in \{2,3,4\}$	$\beta \in \{0,1,2\}$
	4	$m \in \{2,3,4\}$	
	5	$m \in \{2,3\}$	
	6	$m = 2$	
Small-to-medium problem (α_2)	$n \in \{6,7,8,9,10,11\}$	$m \in \{2,3,4,5\}$	$\beta \in \{0,1,2\}$

$$T_{4\_min}(k) = \begin{cases} \max_{1 \le j \le n} \left\{ \sum_{i=k+1}^{m} p_{ji} \right\} & k < m \\ 0 & k = m. \end{cases}$$

Next, combining the earliest completion time and estimated latest finish time, i.e., $B_{4\_E}(k)$ and $B_{4\_L}(k)$, we have the following as a makespan lower bound.

$$LB_4(k) = \max \left(B_{4\_E}(k) + T_{4\_max}(k), B_{4\_L}(k) + T_{4\_min}(k) \right)$$

Example: Consider the example in Table 1. The optimal makespan of the problem is 269, as obtained by a mathematical programming model. Table 5 shows the obtained lower bound values.

As shown in Table 5, LB_3 and LB_4 seem to be better indicators of the optimal makespan. For brevity, the methods of computing the lower bound in this section are given in the appendix.

E. THE LOWER BOUND EFFICIENCY

Among the five lower bounds, it was proven by Santos et al. [14] that LB_1 is dominated by LB_2. In this section, we provide two propositions to show that LB_0 and LB_3 are dominated by LB_4. Thus, we only compare LB_2 and LB_4 in the later computational experiments.

Proposition 1: LB_4 dominates LB_0.

Proof: As mentioned above, $LB_0 = \max_{1 \le j \le n} \{\sum_{i=1}^{m} p_{ji}\}$ and $LB_4 = \max_{1 \le j \le n} \{LB_4(k)\}$, where

$$LB_4(k) = \max(B_{4\_E}(k) + T_{4\_max}(k), B_{4\_L}(k) + T_{4\_min}(k))$$

$$B_{4\_E}(k) = \min_{1 \le j \le n} \left\{ \sum_{i=1}^{k} p_{ji} \right\},$$

$$B_{4\_L}(k) = \max_{1 \le j \le n} \left(H_{4\_E}(k) + B_3(k), \max_{1 \le j \le n} \left\{ \sum_{i=1}^{k} p_{ji} \right\} \right).$$

TABLE 7. Comparing the eight models with respect to the average makespan for small problems.

Parameter of machines	Jobs_stages	$Av_{C_{max}}$							
		MIP(N4)	MIP(C)	MIP(KP)	MIP(PD)	MIP(N1)	MIP(N2)	MIP(N3)	MIP(K)
$\beta = 0$	3_2	132.9	132.9	132.9	132.9	132.9	132.9	132.9	132.9
	3_3	208.4	208.4	208.4	208.4	208.4	208.4	208.4	208.4
	3_4	233.3	233.3	233.3	233.3	233.3	233.3	233.3	233.3
	4_2	153.7	153.7	153.7	153.7	153.7	153.7	153.7	153.7
	4_3	207.4	207.4	207.4	207.4	207.4	207.4	207.4	207.4
	4_4	262.7	262.7	262.7	262.7	262.7	262.7	262.7	262.7
	5_2	159.8	159.8	159.8	159.8	159.8	159.8	159.8	159.8
	5_3	192.4	192.4	192.4	192.4	192.4	192.4	192.4	192.4
	6_2	192.4	192.4	192.4	192.4	192.4	192.4	192.4	192.4
$\beta = 1$	3_2	135.1	135.1	135.1	135.1	135.1	135.1	135.1	135.1
	3_3	194.1	194.1	194.1	194.1	194.1	194.1	194.1	194.1
	3_4	249.0	249.0	249.0	249.0	249.0	249.0	249.0	249.0
	4_2	152.7	152.7	152.7	152.7	152.7	152.7	152.7	152.7
	4_3	213.7	213.7	213.7	213.7	213.7	213.7	213.7	213.7
	4_4	232.2	232.2	232.2	232.2	232.2	232.2	232.2	232.2
	5_2	159.4	159.4	159.4	159.4	159.4	159.4	159.4	159.4
	5_3	193.8	193.8	193.8	193.8	193.8	193.8	193.8	193.8
	6_2	162.6	162.6	162.6	162.6	162.6	162.6	162.6	162.6
$\beta = 2$	3_2	191.9	191.9	191.9	191.9	191.9	191.9	191.9	191.9
	3_3	217.7	217.7	217.7	217.7	217.7	217.7	217.7	217.7
	3_4	282.8	282.8	282.8	282.8	282.8	282.8	282.8	282.8
	4_2	202.3	202.3	202.3	202.3	202.3	202.3	202.3	202.3
	4_3	249.8	249.8	249.8	249.8	249.8	249.8	249.8	249.8
	4_4	328.9	328.9	328.9	328.9	328.9	328.9	328.9	328.9
	5_2	247.2	247.2	247.2	247.2	247.2	247.2	247.2	247.2
	5_3	305.0	305.0	305.0	305.0	305.0	305.0	305.0	305.0
	6_2	278.5	278.5	278.5	278.5	278.5	278.5	278.5	278.5

TABLE 8. Comparing the eight models with respect to the average CPU time for small problems.

Parameter of machines	Jobs_stages	Avg_{CPU}							
		MIP(N4)	MIP(C)	MIP(KP)	MIP(PD)	MIP(N1)	MIP(N2)	MIP(N3)	MIP(K)
$\beta = 0$	3_2	0.191	0.031	0.067	0.077	0.053	0.047	0.042	0.042
	3_3	0.030	0.028	0.047	0.039	0.038	0.049	0.047	0.042
	3_4	0.041	0.033	0.049	0.047	0.039	0.053	0.044	0.036
	4_2	0.053	0.044	0.055	0.059	0.055	0.064	0.059	0.042
	4_3	0.056	0.041	0.058	0.063	0.063	0.058	0.094	0.055
	4_4	0.050	0.053	0.055	0.081	0.055	0.070	0.086	0.061
	5_2	0.050	0.056	0.066	0.089	0.072	0.089	0.244	0.059
	5_3	0.075	0.066	0.069	0.134	0.072	0.105	0.286	0.080
	6_2	0.116	0.064	0.150	0.289	0.102	0.136	2.613	0.092
$\beta = 1$	3_2	0.008	0.025	0.019	0.028	0.044	0.045	0.025	0.020
	3_3	0.023	0.033	0.045	0.034	0.036	0.047	0.028	0.034
	3_4	0.019	0.036	0.038	0.030	0.036	0.053	0.034	0.025
	4_2	0.045	0.036	0.058	0.041	0.050	0.059	0.069	0.041
	4_3	0.042	0.048	0.066	0.055	0.058	0.058	0.063	0.056
	4_4	0.050	0.060	0.067	0.062	0.039	0.064	0.098	0.072
	5_2	0.053	0.048	0.069	0.084	0.072	0.063	0.109	0.066
	5_3	0.058	0.066	0.064	0.098	0.072	0.083	0.120	0.081
	6_2	0.073	0.080	0.091	0.183	0.077	0.089	2.325	0.083
$\beta = 2$	3_2	0.023	0.036	0.044	0.041	0.041	0.061	0.052	0.038
	3_3	0.055	0.042	0.041	0.060	0.041	0.053	0.049	0.045
	3_4	0.047	0.050	0.059	0.047	0.048	0.055	0.053	0.049
	4_2	0.052	0.036	0.056	0.045	0.053	0.047	0.067	0.030
	4_3	0.044	0.061	0.061	0.055	0.063	0.064	0.069	0.055
	4_4	0.050	0.058	0.070	0.047	0.064	0.062	0.061	0.059
	5_2	0.049	0.053	0.081	0.072	0.077	0.063	0.070	0.067
	5_3	0.055	0.084	0.167	0.102	0.092	0.091	0.111	0.100
	6_2	0.078	0.066	0.149	0.087	0.120	0.094	0.434	0.083

When

$$k = m, B_{4\_E}(m) = \min_{1\le j\le n}\left\{\sum\nolimits_{i=1}^{m} p_{ji}\right\},$$

$$T_{4\_max}(m) = T_{4\_min}(m) = 0, \text{ and}$$

$$B_{4\_L}(m) = \max_{1\le j\le n}\left(H_{4\_E}(m) + B_3(m),\ \max_{1\le j\le n}\left\{\sum\nolimits_{i=1}^{m} p_{ji}\right\}\right).$$

We can obtain the following:

$$LB_4(m) = \max\left(B_{4\_E}(m) + 0, B_{4\_L}(k) + 0\right)$$

$$= \max\left(\max_{1\le j\le n}\left\{\sum\nolimits_{i=1}^{m} p_{ji}\right\}, \max(H_{4\_E}(m) + B_3(m),\ \max_{1\le j\le n}\left\{\sum\nolimits_{i=1}^{m} p_{ji}\right\})\right) \ge \max_{1\le j\le n}\left\{\sum\nolimits_{i=1}^{m} p_{ji}\right\} = LB_0.$$

TABLE 9. Comparing the eight models with respect to the average makespan for small-to-medium problems.

Parameter of machines	Jobs_stages	$Avg_{C_{max}}$							
		MIP(N4)	MIP(C)	MIP(KP)	MIP(PD)	MIP(N1)	MIP(N2)	MIP(N3)	MIP(K)
$\beta = 0$	6_2	150.0	150.0	150.0	150.0	150.0	150.0	150.0	150.0
	6_3	220.8	220.8	220.8	220.8	220.8	220.8	220.8	220.8
	6_4	265.0	265.0	265.0	265.0	265.0	265.0	265.0	265.0
	6_5	335.2	335.2	335.2	335.2	335.2	335.2	335.2	335.2
	7_2	178.6	178.6	178.6	178.6	178.6	178.6	178.6	178.6
	7_3	263.4	263.4	263.4	263.4	263.4	263.4	263.4	263.4
	7_4	310.0	310.0	310.0	310.0	310.0	310.0	310.0	310.0
	7_5	356.8	356.8	356.8	356.8	356.8	356.8	356.8	356.8
	8_2	241.6	241.6	241.6	241.6	241.6	241.6	241.6	241.6
	8_3	260.0	260.0	260.0	260.0	260.0	260.0	260.0	260.0
	8_4	293.8	293.8	293.8	293.8	293.8	293.8	293.8	293.8
	8_5	359.6	359.6	359.6	359.6	359.6	359.6	359.6	359.6
	9_2	247.8	247.8	247.8	247.8	247.8	247.8	247.8	247.8
	9_3	289.8	289.8	289.8	289.8	289.8	289.8	289.8	289.8
	9_4	324.8	324.8	324.8	324.8	324.8	324.8	**325.8**	324.8
	9_5	387.2	387.2	387.2	387.2	**388.2**	387.2	**388.4**	387.2
	10_2	257.6	257.6	257.6	257.6	257.8	257.6	257.6	257.6
	10_3	331.2	331.2	**331.4**	331.2	331.2	331.2	331.2	331.2
	10_4	344.0	344.0	**344.6**	344.0	344.0	344.0	**346.6**	344.0
	10_5	398.4	398.4	**399.2**	398.4	**398.8**	398.6	**403.8**	398.4
	11_2	299.2	299.2	299.2	299.2	**300.2**	299.2	299.2	299.2
	11_3	341.2	341.2	**344.8**	341.2	**345.6**	341.2	**343.8**	341.2
	11_4	330.6	330.6	**335.4**	330.6	**335.0**	**331.0**	**335.0**	330.6
	11_5	424.4	424.4	**434.2**	**424.6**	**441.6**	**425.6**	**438.8**	424.4
$\beta = 1$	6_2	186.6	186.6	186.6	186.6	186.6	186.6	186.6	186.6
	6_3	225.2	225.2	225.2	225.2	225.2	225.2	225.2	225.2
	6_4	272.8	272.8	272.8	272.8	272.8	272.8	272.8	272.8
	6_5	332.8	332.8	332.8	332.8	332.8	332.8	332.8	332.8
	7_2	172.4	172.4	172.4	172.4	172.4	172.4	172.4	172.4
	7_3	227.8	227.8	227.8	227.8	227.8	227.8	227.8	227.8
	7_4	300.6	300.6	300.6	300.6	300.6	300.6	300.6	300.6
	7_5	343.2	343.2	343.2	343.2	343.2	343.2	343.2	343.2
	8_2	180.6	180.6	180.6	180.6	180.6	180.6	180.6	180.6
	8_3	243.6	243.6	243.6	243.6	243.6	243.6	243.6	243.6
	8_4	322.4	322.4	322.4	322.4	322.4	322.4	322.4	322.4
	8_5	347.4	347.4	347.4	347.4	347.4	347.4	**348.2**	347.4
	9_2	201.0	201.0	201.0	201.0	201.0	201.0	201.0	201.0
	9_3	255.8	255.8	255.8	255.8	255.8	255.8	255.8	255.8
	9_4	310.0	310.0	310.0	310.0	310.0	310.0	310.0	310.0
	9_5	372.2	372.2	372.2	372.2	372.2	**372.8**	**375.6**	372.2
	10_2	212.4	212.4	212.4	212.4	212.4	212.4	212.4	212.4
	10_3	265.4	265.4	265.4	265.4	265.4	265.4	**266.2**	265.4
	10_4	323.6	323.6	**326.8**	323.6	**326.2**	323.6	**326.2**	323.6
	10_5	385.2	385.2	**387.8**	385.2	**385.6**	**385.6**	**388.2**	385.2
	11_2	227.8	227.8	227.8	227.8	**228.2**	227.8	228.2	227.8
	11_3	277.6	277.6	**280.8**	277.6	**280.6**	277.6	**280.4**	277.6
	11_4	334.2	334.2	**341.4**	**334.8**	**340.4**	334.2	**345.6**	334.2
	11_5	**389.0**	388.8	**396.6**	**389.2**	**402.8**	**391.2**	**393.6**	388.8
$\beta = 2$	6_2	225.2	225.2	225.2	225.2	225.2	225.2	225.2	225.2
	6_3	296.6	296.6	296.6	296.6	296.6	296.6	296.6	296.6
	6_4	417.8	417.8	417.8	417.8	417.8	417.8	417.8	417.8
	6_5	471.4	471.4	471.4	471.4	471.4	471.4	471.4	471.4
	7_2	292.2	292.2	292.2	292.2	292.2	292.2	292.2	292.2
	7_3	353.4	353.4	353.4	353.4	353.4	353.4	353.4	353.4
	7_4	403.2	403.2	403.2	403.2	403.2	403.2	403.2	403.2
	7_5	529.8	529.8	529.8	529.8	529.8	529.8	529.8	529.8
	8_2	318.8	318.8	318.8	318.8	318.8	318.8	318.8	318.8
	8_3	381.8	381.8	381.8	381.8	381.8	381.8	381.8	381.8
	8_4	492.8	492.8	492.8	492.8	492.8	492.8	492.8	492.8
	8_5	500.2	500.2	500.4	500.2	500.2	500.2	500.2	500.2
	9_2	448.0	448.0	448.0	448.0	448.0	448.0	448.0	448.0
	9_3	511.8	511.8	511.8	511.8	511.8	511.8	**512.2**	511.8
	9_4	582.4	582.4	582.4	582.4	582.4	582.4	582.4	582.4
	9_5	610.8	610.8	610.8	610.8	**611.4**	610.8	**613.0**	610.8
	10_2	299.0	299.0	**299.2**	299.0	299.0	299.0	299.0	299.0
	10_3	571.4	571.4	571.4	571.4	571.4	571.4	571.4	571.4
	10_4	653.8	653.8	**654.6**	653.8	**654.2**	653.8	653.8	653.8
	10_5	589.0	589.0	**592.2**	589.0	**593.4**	589.0	**596.2**	589.0
	11_2	493.0	493.0	493.0	493.0	**493.2**	493.0	493.0	493.0
	11_3	540.0	540.0	**542.4**	540.0	**541.6**	540.0	**541.6**	540.0
	11_4	585.0	585.0	**587.8**	585.0	**593.6**	585.0	**590.4**	585.0
	11_5	630.4	630.4	**657.4**	630.4	**660.2**	630.4	**632.2**	630.4

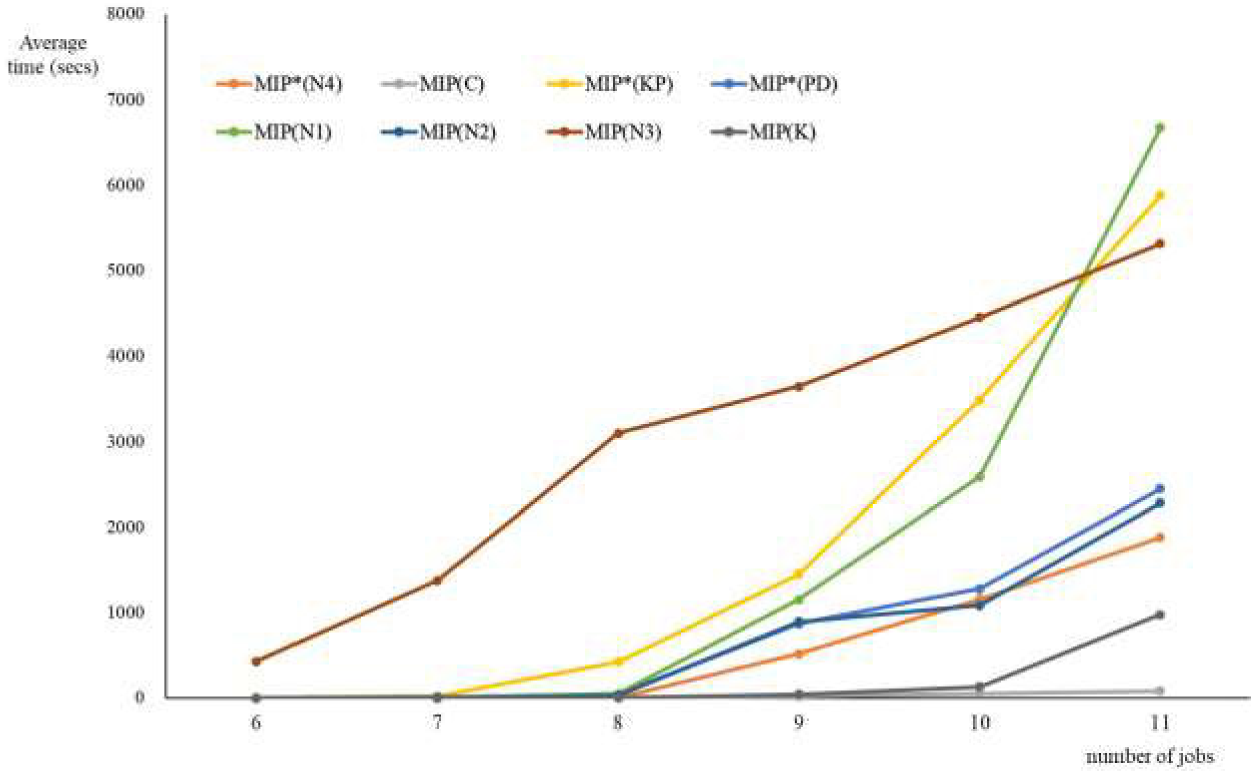

FIGURE 2. The average computation times required by different models.

Proposition 2: LB_4 dominates LB_3.

Proof: $LB_3 = \max_{1\le k\le m} \{LB_3(k)\}$, where

$$LB_3(k) = H_3(k) + B_3(k) + T_3(k),$$

$$H_3(k) = \begin{cases} 0 & k = 1 \\ \min\limits_{1\le j\le n}\left(\sum_{i=1}^{k-1} p_{ji}\right) & k > 1, \end{cases}$$

$$T_3(k) = \begin{cases} \min\limits_{1\le j\le n}\left(\sum_{i=k+1}^{m} p_{ji}\right) & k < m \\ 0 & k = m. \end{cases}$$

For LB_4, $LB_4 = \max_{1\le k\le m} \{LB_4(k)\}$, where

$$LB_4(k) = max(B_{4\_E}(k) + T_{4\_max}(k), B_{4\_L}(k) + T_{4\_min}(k))$$

$$B_{4\_L}(k) = \max_{1\le j\le n}\left(H_{4\_E}(k) + B_3(k), \max_{1\le j\le n}\left\{\sum_{i=1}^{k} p_{ji}\right\}\right)$$

Comparing the two pairs

$$(H_{4\_E}(k), H_3(k)) and (T_{4\_min}(k), T_3(k)),$$

we find that $H_{4\_E}(k) = H_3(k)$ and $T_{4\_min}(k) = T_3(k)$

$$\begin{aligned} LB_4(k) &= max\left(B_{4\_E}(k) + T_{4\_max}(k), B_{4\_L}(k) + T_{4\_min}(k)\right) \\ &= max\left(B_{4\_E}(k) + T_{4\_max}(k), B_{4\_L}(k) + T_3(k)\right) \\ &= max\Big(B_{4\_E}(k) + T_{4\_max}(k), \Big(H_3(k) + B_3(k), \max_{1\le j\le n}\left\{\sum_{i=1}^{k} p_{ji}\right\}\Big) + T_3(k)\Big) \\ &= max(B_{4\_E}(k) + T_{4\_max}(k), \max_{1\le j\le n} (H_3(k) + B_3(k) + T_3(k), \max_{1\le j\le n}\left\{\sum_{i=1}^{k} p_{ji}\right\} + T_3(k)\Big) \\ &= max(B_{4\_E}(k) + T_{4\_max}(k), max\{LB_3(k), \max_{1\le j\le n}\left\{\sum_{i=1}^{k} p_{ji}\right\} + T_3(k)\} \ge LB_3 \end{aligned}$$

V. COMPUTATIONAL EXPERIMENTS

This section shows the computational results of our experimentation in this study. All the experiments were run on a personal computer with an Intel Xeon E-2124 3.4 GHz CPU with 32 GB of DRAM. IBM ILOG CPLEX optimization studio version 12.7.1 was used to formulate all mathematical programming models in this study to obtain the optimal solutions, and the computation time of a model for solving each instance in CPLEX was limited to 7200 s. The programming language C++ in Visual Studio 2020 was used to code all procedures for the lower bounds mentioned above.

A. BENCHMARK INSTANCES

For a fair comparison, we adopted the testbed proposed by Fernandez-Viagas et al. [16]. These test problems include two

TABLE 10. Comparing the eight models with respect to the average computation time for small-to-medium problems.

Parameter of machines	Jobs_stages	Avg_{CPU}							
		MIP(N4)	MIP(C)	MIP(KP)	MIP(PD)	MIP(N1)	MIP(N2)	MIP(N3)	MIP(K)
$\beta = 0$	6_2	0.100	0.087	0.175	0.253	0.100	0.106	0.584	0.075
	6_3	0.103	0.109	0.134	0.466	0.113	0.156	0.525	0.100
	6_4	0.178	0.100	0.319	0.903	0.303	0.516	1477.347	0.150
	6_5	0.234	0.100	0.403	2.444	0.275	1.022	560.069	0.241
	7_2	0.219	0.088	0.441	0.844	0.519	0.259	78.547	0.131
	7_3	0.291	0.110	1.109	1.609	1.212	0.706	3026.909	0.238
	7_4	0.494	0.134	0.975	1.956	0.950	0.813	931.341	0.394
	7_5	0.472	0.259	5.453	103.019	4.959	5.072	2924.106	0.769
	8_2	0.766	0.275	2.003	4.378	2.462	0.722	314.366	1.603
	8_3	3.250	0.613	22.437	9.400	11.256	2.406	4328.510	4.738
	8_4	4.266	0.325	6.613	363.256	10.153	11.619	4378.306	8.138
	8_5	1.625	0.569	135.619	20.291	159.503	16.503	4330.022	1.325
	9_2	6.106	2.753	111.103	51.516	92.725	1.816	1585.691	3.703
	9_3	16.975	21.019	85.015	55.281	1479.753	8.953	3172.537	14.315
	9_4	1454.732	9.887	201.781	1644.831	245.797	529.594	7203.716	168.444
	9_5	2883.738	2.372	2461.281	2887.932	2891.728	2949.003	4322.737	11.232
	10_2	59.172	13.566	325.075	282.428	1819.888	14.419	5816.515	138.125
	10_3	1660.987	267.759	3749.234	323.387	862.253	14.503	4463.831	35.050
	10_4	4089.585	15.934	4994.047	4324.581	3077.988	2939.375	5908.903	147.006
	10_5	2433.813	18.294	3265.800	2401.738	3167.325	4295.541	5764.584	456.337
	11_2	41.531	75.903	5438.466	1574.534	6011.360	31.688	4551.503	118.594
	11_3	1115.406	54.503	7217.997	3324.788	7210.218	2907.553	6961.943	1151.890
	11_4	2954.203	10.690	6178.550	6245.947	5091.994	5769.309	7201.972	75.625
	11_5	4782.741	183.181	5960.997	5763.037	7205.384	4427.578	7200.231	1587.409
$\beta = 1$	6_2	0.085	0.053	0.081	0.484	0.134	0.137	3.625	0.088
	6_3	0.134	0.084	0.162	0.709	0.156	0.219	189.150	0.103
	6_4	0.122	0.078	0.159	0.475	0.106	0.322	0.491	0.125
	6_5	0.350	0.125	0.456	1.641	0.300	1.591	2882.019	0.209
	7_2	0.178	0.109	0.162	0.584	0.137	0.238	422.250	0.109
	7_3	0.300	0.107	0.447	2.237	0.347	1.347	1934.038	0.150
	7_4	0.284	0.116	0.556	58.491	0.566	9.722	1440.544	0.231
	7_5	0.472	0.259	5.453	103.019	4.959	5.072	2924.106	0.769
	8_2	0.187	0.094	1.088	1.056	0.712	0.356	3657.472	0.191
	8_3	1.991	0.175	3.028	1.528	4.672	2.294	7202.769	0.547
	8_4	3.244	0.541	5.128	7.662	4.853	36.609	4322.300	2.347
	8_5	3.416	0.878	10.841	8.409	15.612	319.713	4324.562	8.516
	9_2	0.878	0.156	17.097	2.303	21.944	1.225	5788.056	0.638
	9_3	8.878	0.784	63.341	62.503	102.256	40.984	7204.531	5.497
	9_4	2.875	0.413	12.628	1.594	16.516	79.959	2918.609	1.069
	9_5	369.969	8.294	848.734	4266.687	272.950	5636.550	7202.041	151.662
	10_2	2.178	0.709	396.994	22.694	438.391	7.109	7208.913	3.085
	10_3	1543.181	6.747	1361.712	62.622	730.059	638.472	7213.306	62.303
	10_4	4089.585	15.934	4994.047	4324.581	3077.988	2939.375	5908.903	147.006
	10_5	640.319	39.872	3047.594	2004.988	1704.928	1854.897	3088.244	118.069
	11_2	8.616	4.538	884.293	160.812	4566.015	33.087	7217.678	246.466
	11_3	106.294	8.822	5027.828	127.832	6983.653	1748.594	7204.031	157.644
	11_4	5810.478	96.897	5349.750	4538.678	7213.690	3403.110	7200.340	1482.519
	11_5	6017.981	28.787	6116.312	5813.084	7212.053	7200.422	7200.241	1061.472
$\beta = 2$	6_2	0.110	0.097	0.212	0.428	0.103	0.137	0.425	0.138
	6_3	0.147	0.097	0.862	0.747	0.369	0.181	15.694	0.222
	6_4	0.153	0.144	2.191	0.697	0.741	0.353	3.350	0.313
	6_5	0.175	0.156	19.637	0.503	1.859	0.632	1.265	0.316
	7_2	0.159	0.181	1.025	0.666	0.500	0.319	1440.488	0.260
	7_3	0.163	0.319	12.978	0.878	1.669	0.434	0.475	0.419
	7_4	0.469	0.272	39.684	0.884	3.128	0.781	2.503	0.691
	7_5	0.237	0.678	134.050	1.487	33.050	1.238	1.619	0.900
	8_2	0.263	0.190	1.975	0.906	1.619	0.381	1440.609	0.397
	8_3	0.953	0.494	1609.581	1.797	53.794	1.269	1440.719	2.097
	8_4	1.128	1.007	308.606	19.237	194.666	1.594	1442.309	2.344
	8_5	0.791	1.191	2955.544	1.431	123.119	2.415	12.294	2.147
	9_2	2.697	1.669	61.909	2.944	63.578	1.919	1482.356	3.716
	9_3	14.528	8.356	2935.581	19.725	1576.638	6.263	1444.075	118.156
	9_4	1.187	135.135	5798.175	1.800	1201.059	5.522	7.741	23.615
	9_5	1441.575	5.050	4798.347	1447.609	5880.519	1448.447	1460.659	29.819
	10_2	20.084	14.937	1669.044	90.078	1043.922	10.059	3209.225	15.309
	10_3	13.603	19.163	5869.862	1444.947	1358.078	19.709	1441.210	31.169
	10_4	43.738	39.244	7211.909	66.785	7203.028	154.285	1471.178	153.253
	10_5	365.600	109.909	7211.475	1281.603	7207.725	1468.184	1456.606	365.437
	11_2	85.416	85.031	6921.719	139.447	6993.294	101.534	491.785	169.513
	11_3	1294.250	232.716	7207.366	789.453	7200.672	470.375	2931.641	2893.150
	11_4	41.850	53.372	7220.753	1506.297	7216.682	208.356	1456.259	2202.363
	11_5	43.569	115.972	7212.203	126.760	7208.616	318.903	4355.206	362.197

TABLE 11. The results of LB_2 and LB_4 compared with the optimal solutions for small-sized problems.

n	m	β	LB_2		LB_4		Improvement % byLB_4		
			Ave. RE (%)	#OP	Ave. RE (%)	#OP	min	average	max
3	2	0	18.27	0	1.64	8	-1.20	25.02	91.40
		1	20.91	0	0.00	10	3.74	29.07	62.39
		2	7.06	5	1.16	8	0.00	7.30	35.43
3	3	0	32.57	0	0.00	10	26.17	50.37	76.54
		1	15.87	0	0.00	10	2.07	19.94	39.89
		2	8.25	3	1.65	8	0.00	7.70	25.74
3	4	0	15.91	0	0.33	9	-1.88	20.92	50.77
		1	19.88	0	0.00	10	12.50	25.79	50.34
		2	11.27	2	2.93	5	0.00	12.33	74.34
4	2	0	17.10	0	4.08	4	-5.95	21.82	101.27
		1	28.46	0	0.00	10	12.31	43.84	86.90
		2	5.42	4	2.57	6	0.00	3.37	22.54
4	3	0	22.22	0	0.98	8	-3.68	32.12	80.65
		1	23.63	0	0.00	10	13.19	32.56	52.67
		2	9.44	3	3.38	6	0.00	7.58	28.91
4	4	0	13.31	0	0.98	8	-0.99	15.39	47.22
		1	14.79	0	0.63	9	0.45	18.01	37.37
		2	9.80	3	2.99	6	0.00	8.83	33.68
5	2	0	9.76	0	7.18	3	-8.97	3.08	15.17
		1	18.44	0	0.92	8	9.33	22.63	53.75
		2	4.00	7	3.14	8	-8.54	1.11	19.63
5	3	0	18.96	0	3.92	5	-6.45	21.60	69.60
		1	16.35	0	1.12	7	-0.69	19.68	45.13
		2	3.04	7	0.88	9	0.00	2.43	13.39
6	2	0	6.09	2	6.96	0	-6.28	-0.34	24.06
		1	14.55	0	6.03	4	-6.04	11.53	52.46
		2	14.14	3	1.96	7	-0.68	18.31	69.16
Global average			14.80	14.4%	2.05	72.6%	-8.97	17.85	101.27

TABLE 12. Nonparametric mann whitney test on LB_2 with LB_4 for small problems.

Hypothesis	LB_2			LB_4			Mann–Whitney U	Significant?
	min	average	max	min	average	max		
LB_2=LB_4 ($\beta = 0$)	0.00	17.13	50.31	0.00	2.90	19.00	5086	Y
LB_2=LB_4 ($\beta = 1$)	2.03	19.21	46.50	0.00	0.97	19.35	4235	Y
LB_2=LB_4 ($\beta = 2$)	0.00	8.05	42.64	0.00	2.30	24.14	7889	Y

data sets: small jobs (α_1) and small-to-medium jobs (α_2). For the number of machines in each stage selected, we used the parameter $\beta$for different machine settings as follows:

- $\beta = 0$ indicates that there are three machines in each stage, except in the single-stage case, where only two machines are available.
- $\beta = 1$ indicates three machines in each stage.
- $\beta = 2$ indicates that a random number of machines for each stage is generated from the range (1, 3).

The parameter set of each problem is given in Table 6, and for each combination of these parameters, ten and five instances are generated for the α_1 and α_2 problem sets, respectively; thus, there are a total of 270 and 360 instances for the α_1 and α_2 problem sets, respectively.

For all instances, the processing times of the jobs in each stage are generated uniformly in the interval [1, 99].

B. COMPARISON RESULTS OF THE MIP MODELS

This subsection compares the proposed MIP model with the other models mentioned above. As mentioned above, all mathematical models are coded in ILOG CPLEX optimization studio version 12.7.1, and the maximum elapsed running time is set as 7200 s. Tables 7 and 8 list the average makespan and computation time of each model over the 10 instances of each small-size problem. From Tables 7 and 8, it is obvious that all models can obtain optimal makespan solutions within a short time. For small-to-medium problems, the average makespan and computation time obtained by each model over the five instances are listed in Tables 9 and 10. For small-to-medium problems, some instances cannot be solved optimally by the models in the limited time. These values are represented in bold in Table 9. The corresponding makespans and elapsed running times obtained by the eight models for each instance in this experiment can be found at the website https://drive.google.com/drive/folders/1nnPVS8FG37q1rX2YjNQtuOa9pIiVWXv_?usp=share_link. In addition, as expected, the average computation times of MIP(C), MIP(K), and MIP∗(N4) are shorter, and it is worth noting that the average computation time of MIP(C) does not increase dramatically as the number of jobs increases, as shown in Fig. 2.

TABLE 13. The results of LB_2 and LB_4 compared with the optimal solutions for small-to-medium problems.

n	m	β	LB_2		LB_4		Improvement % by LB_4		
			Ave. RE (%)	#OP	Ave. RE (%)	#OP	min	average	max
6	2	0	5.36	0.00	5.51	1.00	-2.72	-0.15	5.26
		1	11.90	0.00	10.43	0.00	-7.43	1.78	11.66
		2	1.99	2.00	4.75	1.00	-4.90	-2.82	0.00
6	3	0	10.33	0.00	7.48	1.00	-4.24	3.41	10.09
		1	15.04	0.00	4.27	2.00	0.00	13.40	34.48
		2	10.21	0.00	7.74	1.00	-4.74	3.07	21.12
6	4	0	8.80	0.00	8.58	1.00	-6.83	0.43	6.28
		1	15.59	0.00	0.41	4.00	4.37	18.43	26.50
		2	3.89	1.00	4.32	1.00	-2.33	-0.47	0.00
6	5	0	14.91	0.00	3.91	2.00	1.65	13.55	27.65
		1	19.59	0.00	1.02	3.00	11.74	23.91	38.71
		2	6.36	0.00	6.36	0.00	0.00	0.00	0.00
7	2	0	5.75	0.00	8.48	0.00	-8.08	-2.77	0.57
		1	9.88	0.00	6.01	1.00	-1.88	4.46	16.33
		2	3.49	3.00	3.68	3.00	-5.05	-0.11	4.52
7	3	0	8.85	0.00	6.14	0.00	-3.41	3.26	15.11
		1	16.94	0.00	4.61	2.00	-3.59	15.36	26.84
		2	5.13	4.00	0.00	5.00	0.00	6.90	34.48
7	4	0	11.65	0.00	7.25	0.00	-9.27	6.11	20.60
		1	19.17	0.00	0.14	4.00	11.30	25.14	54.64
		2	3.93	1.00	3.93	1.00	0.00	0.00	0.00
7	5	0	14.10	0.00	2.74	2.00	-2.00	14.12	24.71
		1	10.95	0.00	1.76	2.00	1.02	10.51	19.31
		2	0.99	4.00	0.99	4.00	0.00	0.00	0.00
8	2	0	2.70	0.00	3.76	0.00	-2.57	-1.09	-0.35
		1	9.45	0.00	10.26	1.00	-3.80	-0.99	2.03
		2	2.27	2.00	2.14	3.00	-6.67	0.33	10.74
8	3	0	1.30	2.00	3.63	0.00	-4.98	-2.36	-0.62
		1	12.17	0.00	9.05	0.00	-6.61	4.00	19.21
		2	2.14	4.00	2.68	4.00	-3.00	-0.60	0.00
8	4	0	5.18	0.00	5.06	0.00	-3.13	0.25	8.05
		1	14.96	0.00	4.31	2.00	-2.89	13.45	31.08
		2	0.92	4.00	0.92	4.00	0.00	0.00	0.00
8	5	0	16.56	1.00	4.74	1.00	-1.39	16.67	49.57
		1	14.14	0.00	6.18	2.00	-6.76	11.05	42.55
		2	2.21	2.00	0.52	3.00	0.00	1.85	9.23
9	2	0	3.30	2.00	5.11	0.00	-3.65	-1.81	0.00
		1	7.99	0.00	9.06	0.00	-4.12	-1.20	4.17
		2	0.82	2.00	1.38	2.00	-1.64	-0.57	0.00
9	3	0	7.25	0.00	7.10	0.00	-6.44	0.81	21.50
		1	9.24	0.00	12.08	0.00	-6.19	-3.13	-0.90
		2	3.19	4.00	3.34	4.00	-0.92	-0.18	0.00
9	4	0	9.28	0.00	6.52	0.00	-3.68	3.32	12.08
		1	22.57	0.00	3.23	3.00	6.30	26.38	44.81
		2	0.34	3.00	0.34	3.00	0.00	0.00	0.00
9	5	0	16.98	0.00	5.20	2.00	3.89	14.35	20.30
		1	9.70	0.00	9.61	0.00	-5.18	0.21	8.12
		2	5.45	0.00	5.80	0.00	-1.95	-0.39	0.00
10	2	0	2.34	1.00	4.33	0.00	-5.04	-2.07	-0.35
		1	8.83	0.00	10.62	0.00	-9.43	-1.81	8.78
		2	1.26	2.00	2.40	1.00	-3.57	-1.15	1.27
10	3	0	2.17	0.00	5.04	0.00	-5.28	-2.94	-1.02
		1	6.13	0.00	10.72	0.00	-8.43	-4.87	-2.84
		2	0.86	4.00	0.86	4.00	0.00	0.00	0.00
10	4	0	4.25	1.00	7.11	0.00	-7.65	-2.89	1.77
		1	11.80	0.00	8.94	1.00	-5.10	3.73	22.04
		2	1.15	3.00	1.15	3.00	0.00	0.00	0.00
10	5	0	5.46	0.00	6.94	1.00	-6.42	-1.53	2.10
		1	10.24	0.00	5.09	2.00	-3.78	6.05	14.66
		2	3.86	4.00	1.52	4.00	0.00	2.89	14.47
11	2	0	0.60	2.00	1.85	0.00	-3.45	-1.26	0.00
		1	3.66	0.00	6.17	0.00	-3.64	-2.60	-0.93
		2	0.28	4.00	0.81	3.00	-2.63	-0.53	0.00
11	3	0	4.12	1.00	6.28	0.00	-3.82	-2.23	-0.36
		1	8.08	0.00	13.90	0.00	-9.86	-6.33	-0.36
		2	0.99	3.00	2.37	2.00	-4.23	-1.41	0.00
11	4	0	8.54	0.00	11.40	0.00	-5.11	-3.15	-1.38
		1	6.62	0.00	11.03	0.00	-8.31	-4.72	0.00
		2	1.09	2.00	1.68	1.00	-2.95	-0.59	0.00
11	5	0	7.94	0.00	5.38	1.00	-6.44	3.13	16.67
		1	11.78	0.00	9.22	0.00	-6.34	3.76	22.70
		2	4.96	1.00	4.91	1.00	0.00	0.06	0.29
Mean			7.39	19.2%	5.17	27.5%	-2.96	3.02	10.80

TABLE 14. Nonparametric Mann Whitney test on LB_2 with LB_4 for small-to-medium problems.

Hypothesis	LB_2			LB_4			Mann–Whitney U	Significant?
	min	average	max	min	average	max		
LB_2=LB_4 ($\beta = 0$)	0.00	7.41	33.14	0.00	5.80	19.45	6836	Y
LB_2=LB_4 ($\beta = 1$)	0.52	11.93	35.33	0.00	7.00	20.88	4233	Y
LB_2=LB_4 ($\beta = 2$)	0.00	2.82	25.64	0.00	2.69	17.35	9030	Y

TABLE 15. Performance of the lower bounds for small problems.

	LB_2	LB_4	$LB^* = \max(LB_2, LB_4)$
Max (%)	30.37	90.00	100
Ave. RE (%)	14.80	2.05	0.02

TABLE 16. Performance of the lower bounds for small-to-medium problems.

	LB_2	LB_4	$LB^* = \max(LB_2, LB_4)$
Max (%)	69.44	57.50	100
Ave. RE (%)	7.39	5.17	3.87

C. COMPARISON OF LOWER BOUNDS AND OPTIMAL SOLUTIONS

1) SMALL PROBLEMS

The problem set (α_1) consists of a group of 270 instances that were mentioned in Section IV-A. Since they are small problems, the corresponding optimal solutions could be obtained using the aforementioned MIP model and thus provide an excellent base for comparison. We use the relative errors of the lower bound values with respect to the optimal solutions, which are calculated as follows:

$$\text{Relative error (RE)} = \frac{Optimal - Lowerbound}{Optimal} \times 100\%$$

Table 11 provides the results for LB_2 and LB_4. The columns under "Ave. RE (%)" give the average RE values over 10 instances. The other columns, under "#OP", denote the number of times each lower bound is equal to the optimal solution. In addition, the columns under "Improvement (%)" give the improvement percentage of LB_4 over LB_2, which is calculated as follows:

$$\text{Improvement (\%)} = \frac{LB_4 - LB_2}{LB_2} \times 100\%$$

As shown in Table 11, 196 of the 270 (72.6%) lower bounds generated by LB_4 accurately predicted the optimal solutions, which was a greater percentage than the 39 of 270 (14.4%) lower bounds generated by LB_2. The mean relative error of LB_4 was 2.05%, much smaller than the 14.80% error of LB_2, indicating that LB_4 can provide an accurate prediction of the optimal solutions. From the column "Improvement (%)", it is seen that LB_4 can improve the accuracy of the lower bounds obtained by LB_2. However, LB_2 is not dominated by LB_4 since in some cases, LB_2 is better than LB_4. We also conduct a nonparametric Mann-Whitney test, in which the null hypothesis H_0 is that $\mu_{LB_2} = \mu_{LB_4}$ and the alternative hypothesis H_1 is that $\mu_{LB_2} \neq \mu_{LB_4}$ with a 0.05 significance level. Table 12 shows that the hypothesis $\mu_{LB_2} = \mu_{LB_4}$ is rejected in each case of $\beta = 0$, $\beta = 1$, and $\beta = 2$.

2) SMALL-TO-MEDIUM PROBLEMS

Problem set α_2 consists of a group of 360 instances that were mentioned in Section IV-A. All optimal solutions of these instances can be obtained by MIP(C) within 7200 s. The Ave. RE, #OP, and improvement (%) results for all problem configurations are found in Table 13. The total average RE (%) values obtained by LB_2 and LB_4 are 7.39% and 5.17%, respectively. On the other hand, LB_4 accurately predicts the optimal solutions in 99 instances among the 360 total instances, which is 27.5% higher than the 19.2% obtained by LB_2. We also applied a nonparametric Mann-Whitney test for the small-to-medium problems, and the results are shown in Table 14. In all cases (β =0, 1, 2), the hypothesis $LB_2 = LB_4$ was rejected with a 0.05 significance level.

We also report, for each of the two lower bounds, the percentage of times the maximum value was obtained among all instances of each experiment (Max (%)) and the average RE, as shown in Tables 15 and 16. From Tables 15 and 16, we draw the following conclusions:

- The lower bound LB_4 is very competitive for small problems since the bound value is more accurate in predicting the optimal value.
- As the size of the problem increases, the performance difference between LB_4 and LB_2 becomes less remarkable.
- A very effective lower bound that is obtained is $LB^* = max(LB_2, LB_4)$. The average REs of LB^* are 0.02% and 3.87%, respectively, in Tables 15 and 16. These two values imply that LB^*, on average, reaches 98% and 96.13% optimality in the two experiments.
- The computation times of LB_2 and LB_4 can be neglected since the lower bound value of each instance is obtained by LB_2 or LB_4 within 0.001 s.

VI. CONCLUSION

This paper concerns a problem that is encountered in modern manufacturing and production systems, that of minimizing the makespan in a hybrid flow shop configuration. Due to its practical relevance and its NP-hard nature, many exact methods and heuristic algorithms have been proposed to solve the considered problem, and some of these methods, including *B&B*, heuristic, and metaheuristic algorithms, have been compared with each other to evaluate their performance, while MIP models have been excluded. This study fills this gap by proposing a new MIP model and comparing it against the existing models in the literature. The results show that the proposed MIP model is competitive in terms of the number of binary variables, the number of continuous variables, and the number of constraints, which is beneficial for decreasing the computational burden of the MIP model, as shown in the experimental results. Consequently, the proposed MIP model can optimally solve each of the 360 instances of small-to-medium problems efficiently.

Another contribution of this paper is that it surveys the existing lower bound procedures and proposes two new lower bounds based on problems of parallel machines with heads and tails and bin-packing problems. Based on the dominance analysis, the proposed lower bound (LB_4) cannot dominate the LB_2 of Santos et al. [37], although LB_4 was significantly better than LB_2 in the experimental comparisons. Therefore, a composite lower bound of $max(LB_2, LB_4)$ is suggested in this paper as a strong indicator to evaluate the distances between the solutions obtained by heuristic algorithms and the optimal solution. The rest of the lower bound methods were dominated by LB_2 or LB_4.

The literature review showed that some mathematical programming formulations have been proposed to solve more complicated HFSPs [44, 10, 41]; thus, this research can be extended to complicated HFSPs considering real production environments and can be used to develop different mathematical programming formulations for comparison or to propose metaheuristic algorithms to obtain good solutions efficiently for variants of FSPs.

APPENDIX A

- **The first lower bound method**

$LB_0 = \max_{1\le j\le n} \{\sum_{k=1}^{m} p_{jk}\}$.

For $j = 1$, $\sum_{k=1}^{4} p_{1k} = (66 + 53 + 20 + 87) = 226$.
For $j = 2$, $\sum_{k=1}^{4} p_{2k} = (37 + 81 + 40 + 92) = 250$.
For $j = 3$, $\sum_{k=1}^{4} p_{3k} = (54 + 48 + 37 + 97) = 236$.
For $j = 4$, $\sum_{k=1}^{4} p_{4k} = (52 + 81 + 23 + 43) = 119$.
Thus, $LB_0 = max\,(226, 250, 236, 110) = 250$.

- **The second lower bound method**

$LB_1 = \max_{1\le j\le n} \{LB_1(k)\}$, where $LB_1(k) = H_1(k) + B_1(k) + T_1(k)$;

$$H_1(k) = \begin{cases} 0 & k = 1 \\ \min_{1\le j\le n} \left\{\sum_{i=1}^{k-1} p_{ji}\right\} & k > 1; \end{cases}$$

$$B_1(k) = \left\lceil \frac{1}{M_k} \left(\sum_{j=1}^{n} p_{jk}\right) \right\rceil;$$

$$T_1(k) = \begin{cases} \min_{1\le j\le n} \left\{\sum_{i=k+1}^{m} p_{jk}\right\} & k < m \\ 0 & k = m \end{cases}$$

When $k = 1$,

$$H_1(1) = 0;\ B_1(1) = \left\lceil \frac{1}{M_1}\left(\sum_{j=1}^{4} p_{j1}\right)\right\rceil = \left\lceil \frac{1}{3}(66 + 37 + 54 + 52)\right\rceil = 70;$$

$$T_1(1) = \min_{1\le j\le n} \left\{\sum_{i=2}^{m} p_{ji}\right\}$$

For $j = 1$, $\sum_{i=2}^{4} p_{1i} = (53 + 20 + 87) = 160$.
For $j = 2$, $\sum_{i=2}^{4} p_{2i} = (81 + 40 + 92) = 213$.
For $j = 3$, $\sum_{i=2}^{4} p_{3i} = (48 + 37 + 97) = 182$.
For $j = 4$, $\sum_{i=2}^{4} p_{4i} = (81 + 23 + 43) = 147$.
Thus, $T_1(1) = min\,(160, 213, 182, 147) = 147$, and $LB_1(1) = H_1(1) + B_1(1) + T_1(1) = (0 + 70 + 147) = 217$.

When $k = 2$,

$$H_1(2) = \min_{1\le j\le n} \left\{\sum_{i=1}^{1} p_{ji}\right\}$$

For $j = 1$, $\sum_{i=1}^{1} p_{1i} = 66$; for $j = 2$, $\sum_{i=1}^{1} p_{2i} = 37$; for $j = 3$, $\sum_{i=1}^{1} p_{3i} = 54$; and for $j = 4$, $\sum_{i=1}^{1} p_{4i} = 52$.
Thus, $H_1(2) = min\,(66, 37, 54, 52) = 37$.

$$B_1(2) = \left\lceil \frac{1}{M_2}\left(\sum_{j=1}^{4} p_{j2}\right)\right\rceil = \left\lceil \frac{1}{3}(53 + 81 + 48 + 81)\right\rceil = 88;$$

$$T_1(2) = \min_{1\le j\le n} \left\{\sum_{i=3}^{4} p_{ji}\right\} = min\,(107, 132, 134, 66) = 66.$$

Thus,

$$LB_1(2) = H_1(2) + B_1(2) + T_1(2) = (37 + 88 + 66) = 191$$

When $k = 3$,

$$H_1(3) = \min_{1\le j\le n} \left\{\sum_{i=1}^{2} p_{ji}\right\} = min\,(119, 118, 102, 133) = 102$$

$$B_1(3) = \left\lceil \frac{1}{M_3}\left(\sum_{j=1}^{4} p_{j3}\right)\right\rceil = \left\lceil \frac{1}{3}(20 + 40 + 37 + 23)\right\rceil = 40;$$

$$T_1(3) = \min_{1\le j\le n} \left\{\sum_{i=4}^{4} p_{ji}\right\} = min\,(87, 92, 97, 43) = 43.$$

Thus,

$$LB_1(3) = H_1(3) + B_1(3) + T_1(3) = (102 + 40 + 43) = 185$$

When $k = 4$,

$$H_1(4) = \min_{1\le j\le n} \left\{\sum_{i=1}^{3} p_{ji}\right\} = min\,(139, 158, 139, 156) = 139$$

$$B_1(4) = \left\lceil \frac{1}{M_4}\left(\sum_{j=1}^{4} p_{j4}\right)\right\rceil = \left\lceil \frac{1}{3}(87 + 92 + 97 + 43)\right\rceil = 107;$$

$T_1(4) = 0$.

Thus,

$$LB_1(4) = H_1(4) + B_1(4) + T_1(4) = (139 + 107 + 0) = 246$$

$$LB_1 = \max_{1\le j\le n} \{LB_1(k)\} = max\,(217, 191, 185, 246) = 246$$

- **The third lower bound method**

$LB_2 = \{LB_2(k)\}$, where

$LB_2(k) = \left\lceil \frac{1}{M_k}(H_2(k) + B_2(k) + T_2(k)) \right\rceil$.
$H_2(k) = \begin{cases} 0 & k = 1 \\ \sum_{y=1}{}_k^M LSA_y(k) & k > 1 \end{cases}$, where $LSA_y(k)$ is sequenced in increasing order of $LS_j(k)$. $LS_j(k) = \sum_{i=1}^{k-1} p_{ji}$.

$$B_2(k) = \sum_{j=1}^{n} p_{jk};$$

$$T_2(k) = \begin{cases} \sum_{y=1}^{M_k} RSA_y(k) & k < m \\ 0 & k = m, \end{cases}$$

where $RSA_y(k)$ is sequenced in increasing order of $RS_j(k)$. $RS_j(k) = \sum_{i=k+1}^{m} p_{ji}$.

When $k = 1$,

$$H_2(1) = 0; B_2(1) = \sum_{j=1}^{4} p_{jk} = (66 + 37 + 54 + 52) = 209$$

$$T_2(1) = \begin{cases} \sum_{y=1}^{3} RSA_y(1) & k < m \\ 0 & k = m \end{cases}$$

When $j = 1$, $RS_1(1) = \sum_{i=2}^{4} p_{1i} = (53 + 20 + 87) = 160 = RSA_2(1)$.
When $j = 2$, $RS_2(1) = \sum_{i=2}^{4} p_{2i} = (81 + 40 + 92) = 213 = RSA_4(1)$.
When $j = 3$, $RS_3(1) = \sum_{i=2}^{4} p_{3i} = (48 + 37 + 97) = 182 = RSA_3(1)$.
When $j = 4$, $RS_4(1) = \sum_{i=2}^{4} p_{4i} = (81 + 23 + 43) = 147 = RSA_1(1)$.

$$T_2(1) = \sum_{y=1}^{3} RSA_y(1) = (147 + 160 + 182) = 489$$

Thus,

$$LB_2(1) = \left\lceil \frac{1}{3}(H_2(1) + B_2(1) + T_2(1)) \right\rceil = \left\lceil \frac{1}{3}(0 + 209 + 489) \right\rceil = 233$$

When $k = 2$,
$H_2(2) = \sum_{y=1}^{3} LSA_y(2)$; $LS_j(2) = \sum_{i=1}^{1} p_{ji}$.
When $j = 1$, $LS_1(2) = \sum_{i=1}^{1} p_{1i} = 66 = LSA_4(2)$.
When $j = 2$, $LS_2(2) = \sum_{i=1}^{1} p_{2i} = 37 = LSA_1(2)$.
When $j = 3$, $LS_3(2) = \sum_{i=1}^{1} p_{3i} = 54 = LSA_3(2)$.
When $j = 4$, $LS_4(2) = \sum_{i=1}^{1} p_{4i} = 52 = RSA_2(2)$.

$$H_2(2) = \sum_{y=1}^{3} LSA_y(2) = (37 + 52 + 54) = 143$$

$$B_2(2) = \sum_{j=1}^{4} p_{j2} = (53 + 81 + 48 + 81) = 263$$

$$T_2(2) = \sum_{y=1}^{3} RSA_y(2)$$

When $j = 1$, $RS_1(2) = \sum_{i=3}^{4} p_{1i} = (20 + 87) = 107 = RSA_2(2)$.
When $j = 2$, $RS_2(2) = \sum_{i=3}^{4} p_{2i} = (40 + 92) = 132 = RSA_3(2)$.
When $j = 3$, $RS_3(2) = \sum_{i=3}^{4} p_{3i} = (37 + 97) = 134 = RSA_4(2)$.
When $j = 4$, $RS_4(2) = \sum_{i=3}^{4} p_{4i} = (23 + 43) = 66 = RSA_1(2)$.

$$T_2(2) = \sum_{y=1}^{3} RSA_y(2) = (66 + 107 + 132) = 305$$

Thus,

$$LB_2(2) = \left\lceil \frac{1}{3}(H_2(2) + B_2(2) + T_2(2)) \right\rceil = \left\lceil \frac{1}{3}(143 + 263 + 305) \right\rceil = 237$$

When $k = 3$,

$$H_2(3) = \sum_{y=1}^{3} LSA_y(3); LS_j(3) = \sum_{i=1}^{2} p_{ji}$$

$LSA_1(3) = 102$; $LSA_2(3) = 118$; $LSA_3(3) = 119$; $LSA_4(3) = 133$

$$H_2(3) = \sum_{y=1}^{3} LSA_y(3) = (102 + 118 + 119) = 339$$

$$B_2(2) = \sum_{j=1}^{4} p_{j3} = (20 + 40 + 37 + 23) = 120$$

$$T_2(3) = \sum_{y=1}^{3} RSA_y(3); RS_j(3) = \sum_{i=4}^{4} p_{ji}$$

$RSA_1(3) = 43$; $RSA_2(3) = 87$; $RSA_3(3) = 92$; $RSA_4(3) = 97$

$$T_2(3) = \sum_{y=1}^{3} RSA_y(3) = (43 + 87 + 92) = 222$$

$$LB_2(3) = \left\lceil \frac{1}{3}(H_2(3) + B_2(3) + T_2(3)) \right\rceil = \left\lceil \frac{1}{3}(339 + 120 + 222) \right\rceil = 227$$

When $k = 4$,

$$H_2(4) = \sum_{y=1}^{3} LSA_y(4); LS_j(4) = \sum_{i=1}^{3} p_{ji}$$

$LSA_1(4) = 139$; $LSA_2(4) = 139$; $LSA_3(4) = 156$; $LSA_4(4) = 158$

$$H_2(4) = \sum_{y=1}^{3} LSA_y(4) = (139 + 139 + 156) = 434$$

$$B_2(4) = \sum_{j=1}^{4} p_{j4} = (87 + 92 + 97 + 43) = 319$$

$$T_2(4) = 0;$$

$$LB_2(4) = \left\lceil \frac{1}{3}(H_2(4) + B_2(4) + T_2(4)) \right\rceil = \left\lceil \frac{1}{3}(434 + 319 + 0) \right\rceil = 251$$

Thus,

$$\boldsymbol{LB_2 = \max_{1 \le j \le n} \{LB_2(k)\} = \max(233, 234, 227, 251) = 251}$$

- **The fourth lower bound method**

$$LB_3 = \{LB_3(k)\},$$

where

$$LB_3(k) = H_3(k) + B_3(k) + T_3(k).$$

When $k = 1$, we have the following.

For $B_3(1)$, we use the following steps:

Step 1. Let

$$BLB_0(2) = \left\lceil \frac{1}{M_1}\sum_{j=1}^{4} p_{j1} \right\rceil = \left\lceil \frac{1}{3}(66+37+54+52) \right\rceil = 70;\ BLB_1(1) = \{p_{j1}\} = max\,(66, 37, 54, 52) = 66.$$

Step 2. Sort the jobs in descending order of p_{jk}, where $\bar{p_1} = p_{11} = 66$, $\bar{p_4} = p_{21} = 37$, $\bar{p_2} = p_{31} = 54$, and $\bar{p_3} = p_{41} = 54$.

Step 3. $M_1 = (n-1) = 3$, $BLB_2(1) = \bar{p_{M_1}} + \bar{p_{M_1+1}} = \bar{p_3} + \bar{p_4} = (52+37) = 89$.

Step 4. Let $L = max\,(BLB_0(1), BLB_1(1), BLB_2(1)) = max\,(70, 66, 89) = 89$.

Step 5. $M_1 = 3$, $(n-2) = 2$, $M_1 > (n-2)$, $\bar{p} = \bar{p_n} = \bar{p_4} = 37$.

Step 6. $J_A = \{y \mid L - \bar{p} < \bar{p_y}\} = \{y \mid (89-37) < \bar{p_y}\} = \{J_1, J_3$.

$$J_B = \left\{y \mid \frac{L}{2} < \bar{p_y} \le L - \bar{p}\right\} = \left\{y \mid \frac{89}{2} < \bar{p_y} \le (89-37)\right\} = \{J_4\}$$

$$J_C = \left\{y \mid \bar{p} \le \bar{p_y} \le \frac{L}{2}\right\} = \left\{y \mid 37 \le \bar{p_y} \le \frac{89}{2}\right\} = \{J_2\}$$

Step 7. Calculate the values of $B_\alpha(L, \bar{p})$ and $B_\beta(L, \bar{p})$.

$$B_\alpha(89, 37) = |J_A| + |J_B| + max\left(0, \left\lceil \frac{\sum_{y\in J_c} \bar{p_y} - \left(L \times |J_B| - \sum_{y\in J_B} \bar{p_y}\right)}{L} \right\rceil\right) = 2 + 1 + max\left(0, \left\lceil \frac{37 - (89 \times 2 - 52)}{89} \right\rceil\right) = 3 + max(0, 0) = 3$$

$$B_\beta(89, 37) = |J_A| + |J_B| + max\left(0, \left\lceil \frac{|J_c| - \sum_{y\in J_B} \left\lfloor \frac{L - \bar{p_y}}{\bar{p}} \right\rfloor}{\left\lfloor \frac{L}{\bar{p}} \right\rfloor} \right\rceil\right) = 2 + 1 + max(\left\lceil \frac{1 - \left\lfloor \frac{89-52}{37} \right\rfloor}{\left\lfloor \frac{89}{37} \right\rfloor} \right\rceil = 3 + max\,(0, 0) = 3$$

Step 8. $B_\alpha(89, 37) = 3$, $B_\beta(89, 37) = 3$, $M_1 = 3$.

Step 9. $B_3(1) = 89$.

For $H_3(1)$ and $T_3(1)$, the calculations are the same as those of $H_1(1)$ and $T_1(1)$; thus,

$$H_3(1) = H_1(1) = 0$$
$$T_3(1) = T_1(1) = 147$$
$$LB_3(1) = H_3(1) + B_3(1) + T_3(1) = (0 + 89 + 147) = 236$$

When $k = 2$, we have the following.

Step 1. Let $BLB_0(2) = \left\lceil \frac{1}{M_2}\sum_{j=1}^{4} p_{j1} \right\rceil = \left\lceil \frac{1}{3}(53+81+48+81) \right\rceil = 88$; $BLB_1(2) = max\,(53, 81, 48, 81) = 81$.

Step 2. Sort the jobs in descending order of p_{j2}, where $\bar{p_1} = p_{22} = 81$, $\bar{p_2} = p_{42} = 81$, $\bar{p_3} = p_{12} = 53$, $\bar{p_4} = p_{32} = 48$.

Step 3. $M_2 = (n-1) = 3$, $BLB_2(2) = \bar{p_{M_2}} + \bar{p_{M_2+1}} = \bar{p_3} + \bar{p_4} = (53+48) = 101$.

Step 4. Let $L = max\,(BLB_0(2), BLB_1(2), BLB_2(2)) = max\,(88, 81, 101) = 101$.

Step 5. $M_2 = 3$, $(n-2) = 2$, $M_2 > (n-2)$, $\bar{p} = \bar{p_n} = \bar{p_4} = 48$.

Step 6. $J_A = \{y \mid L - \bar{p} < \bar{p_y}\} = \{y \mid (101-48) < \bar{p_y}\} = \{J_2, J_4$.

$$J_B = \left\{y \mid \frac{L}{2} < \bar{p_y} \le L - \bar{p}\right\} = \left\{y \mid \frac{101}{2} < \bar{p_y} \le (101-48)\right\} = \{J_1\}$$

$$J_C = \left\{y \mid \bar{p} \le \bar{p_y} \le \frac{L}{2}\right\} = \left\{y \mid 48 \le \bar{p_y} \le \frac{101}{2}\right\} = \{J_3\}$$

Step 7. Calculate the values of $B_\alpha(L, \bar{p})$ and $B_\beta(L, \bar{p})$.

$$B_\alpha(101, 48) = |J_A| + |J_B| + max\left(0, \left\lceil \frac{\sum_{y\in J_c} \bar{p_y} - \left(L \times |J_B| - \sum_{y\in J_B} \bar{p_y}\right)}{L} \right\rceil\right) = 2 + 1 + max\left(0, \left\lceil \frac{48 - (101 \times 1 - 53)}{101} \right\rceil\right) = 3 + max(0, 0) = 3$$

$$B_\beta(101, 48) = |J_A| + |J_B| + max\left(0, \left\lceil \frac{|J_c| - \sum_{y\in J_B} \left\lfloor \frac{L - \bar{p_y}}{\bar{p}} \right\rfloor}{\left\lfloor \frac{L}{\bar{p}} \right\rfloor} \right\rceil\right) = 2 + 1 + max(\left\lceil \frac{1 - \left\lfloor \frac{101-53}{48} \right\rfloor}{\left\lfloor \frac{101}{48} \right\rfloor} \right\rceil = 3 + max\,(0, 0) = 3$$

Step 8. $B_\alpha(101, 48) = 3$, $B_\beta(101, 48) = 3$, $M_2 = 3$.

Step 9. $B_3(2) = 101$.
For $H_3(2)$ and $T_3(2)$, we can obtain the following:

$$H_3(2) = H_1(2) = 37$$
$$T_3(2) = T_1(2) = 66$$

Thus, $LB_3(2) = H_3(2) + B_3(2) + T_3(2) = (37 + 101 + 66) = 204$.

Using the above procedure for the remaining stages, we can obtain the following:

$$\begin{aligned} LB_3(3) &= H_3(3) + B_3(3) + T_3(3) \\ &= (102 + 43 + 43) = 188 \\ LB_3(4) &= H_3(4) + B_3(4) + T_3(4) \\ &= (139 + 130 + 0) = 269 \end{aligned}$$

The result will be

$$LB_3 = \max_{1\le k\le m} \{LB_3(k)\} = \max(236, 204, 188, 269) = 269$$

- **The fifth lower bound method**

$$LB_4 = \max_{1\le k\le m} \{LB_4(k)\},$$

where

$$LB_4(k) = \max(B_{4\_E}(k) + T_{4\_max}(k), B_{4\_L}(k) + T_{4\_min}(k)).$$

For the first two parts,

$$B_{4\_E}(k) = \min_{1\le j\le n} \{\sum_{i=1}^{k} p_{ji}\}, \text{ and } T_{4\_max}(k) = \begin{cases} \min_{1\le j\le n} \{\sum_{i=k+1}^{m} p_{ji}\} & k < m \\ 0 & k = m \end{cases}$$

For the last two parts, i.e., $B_{4\_L}(k)$ and $T_{4\_min}(k)$, $B_{4\_L}(k) = \max(H_{4\_E}(k) + B_3(k), B_{4\_max}(k))$, where $H_{4\_E}(k) = \begin{cases} 0 & k = 1 \\ \min_{1\le j\le n} \sum_{i=1}^{k-1} p_{ji} & k > 1 \end{cases}$; the calculation of $B_3(k)$ uses the same procedure in steps 1 to 7 of the fourth lower bound procedure.

$$B_{4\_max}(k) = \min_{1\le j\le n} \{\sum_{i=1}^{k} p_{ji}\} \text{ and } T_{4\_min}(k) = \begin{cases} \min_{1\le j\le n} \{\sum_{i=k+1}^{m} p_{ji}\} & k < m \\ 0 & k = m \end{cases}$$

When $k = 1$,

$$\begin{aligned} H_{4\_E}(1) = 0,\ B_{4\_E}(1) &= \min_{1\le j\le n} \left\{\sum_{i=1}^{1} p_{ji}\right\} \\ &= \min(66, 37, 54, 52) = 37,\ T_{4\_min}(1) \\ &= \{\sum_{i=2}^{4} p_{ji}\} = \min(160, 213, 182, 147) = 147 \end{aligned}$$

$B_3(1) = 89$, which is obtained by steps 1 to 9 of the fourth lower bound method.

$$\begin{aligned} B_{4\_max}(1) &= \max_{1\le j\le n} \{\sum_{i=1}^{1} p_{ji}\} \\ &= \max(66, 37, 54, 52) = 66 \\ B_{4\_L}(1) &= \max\left(H_{4\_E}(1) + B_3(1), B_{4\_max}(1)\right) \\ &= \max(0 + 89, 66) = 89 \\ T_{4\_max}(1) &= \max_{1\le j\le n} \{\sum_{i=2}^{4} p_{ji}\} \\ &= \max(160, 213, 182, 147) = 213 \\ LB_4(1) &= \max\left(B_{4\_E}(1) + T_{4\_max}(1), B_{4\_L}(1) + T_{4\_min}(1)\right) \\ &= \max((37 + 213), (89 + 147)) = 250 \end{aligned}$$

When $k = 2$,

$$H_{4\_E}(2) = \min_{1\le j\le n} \sum_{i=1}^{1} p_{ji} = \min(66, 37, 54, 52) = 37,$$

$$B_{4\_E}(2) = \min_{1\le j\le n} \left\{\sum_{i=1}^{2} p_{ji}\right\} = \min(119, 118, 102, 133) = 102,$$

$$\begin{aligned} T_{4\_min}(2) &= \min_{1\le j\le n} \{\sum_{i=3}^{4} p_{ji}\} \\ &= \min((20 + 87), (40 + 92,) \\ &\quad (37 + 97), (23 + 43)) = 66 \end{aligned}$$

$B_3(2) = 101$, which is obtained by steps 1 to 9 of the fourth lower bound method.

$$\begin{aligned} B_{4\_max}(2) &= \max_{1\le j\le n} \{\sum_{i=1}^{2} p_{ji}\} \\ &= \max(119, 118, 102, 133) = 133 \\ B_{4\_L}(2) &= \max\left(H_{4\_E}(2) + B_3(2), B_{4\_max}(2)\right) \\ &= \max((37 + 101), 133) = 138 \\ T_{4\_max}(2) &= \max_{1\le j\le n} \{\sum_{i=3}^{4} p_{ji}\} \\ &= \max(107, 132, 134, 66) = 134 \\ LB_4(2) &= \max\left(B_{4\_E}(2) + T_{4\_max}(2), B_{4\_L}(2) + T_{4\_min}(2)\right) \\ &= \max((102 + 134), (138 + 66)) = 236 \end{aligned}$$

Using the above procedure for the remaining stages, we can obtain the following:

$$\begin{aligned} LB_4(3) &= \max(B_{4\_E}(3) + T_{4\_max}(3), B_{4\_L}(3) + T_{4\_min}(3)) \\ &= \max((139 + 97), (158 + 43)) = 236 \\ LB_4(4) &= \max(B_{4\_E}(4) + T_{4\_max}(4), B_{4\_L}(4) + T_{4\_min}(4)) \\ &= \max((199 + 0), (269 + 0)) = 269 \end{aligned}$$

Thus,

$$LB_4 = \max_{1\le k\le m} \{LB_4(k)\} = \max(250, 236, 236, 269) = 269$$

ACKNOWLEDGMENT

The authors would like to thank Victor Fernandez-Viagas, Paz Perez-Gonzalez, and Jose M. Framinan for the benchmark problems.

REFERENCES

[1] H.-X. Qin, Y.-Y. Han, Y.-P. Liu, J.-Q. Li, Q.-K. Pan, and Xue-Han, "A collaborative iterative greedy algorithm for the scheduling of distributed heterogeneous hybrid flow shop with blocking constraints," *Exp. Syst. Appl.*, vol. 201, Sep. 2022, Art. no. 117256.

[2] B. Naderi, R. Ruiz, and M. Zandieh, "Algorithms for a realistic variant of flowhop scheduling," *Comput. Oper. Res.*, vol. 37, pp. 236–246, 2010.

[3] F. S. Şerifoğlu and G. Ulusoy, "Multiprocessor task scheduling in multistage hybrid flow-shops: A genetic algorithm approach," *J. Oper. Res. Soc.*, vol. 55, no. 5, pp. 504–512, May 2004.

[4] C.-J. Liao, E. Tjandradjaja, and T.-P. Chung, "An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem," *Appl. Soft. Comput.*, vol. 12, no. 6, pp. 1755–1764, 2012.

[5] M. K. Marichelvam, T. Prabaharan, and X. S. Yang, "Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan," *Appl. Soft Comput.*, vol. 19, no. 1, pp. 93–101, Jun. 2014.

[6] M. Dios, V. Fernandez-Viagas, and J. M. Framinan, "Efficient heuristics for the hybrid flow shop scheduling problem with missing operations," *Comput. Ind. Eng.*, vol. 115, pp. 88–99, Jan. 2018.

[7] J. Y. Long, Z. Zheng, X. Q. Gao, and P. M. Pardalos, "Scheduling a realistic hybrid flow shop with stage skipping and adjustable processing time in steel plants," *Appl. Soft Comput.*, vol. 64, pp. 536–549, Mar. 2018.

[8] K. Peng, Q.-K. Pan, L. Gao, B. Zhang, and X. Pang, "An improved artificial bee colony algorithm for real-world hybrid flowshop rescheduling in steelmaking-refining-continuous casting process," *Comput. Ind. Eng.*, vol. 122, pp. 235–250, Aug. 2018.

[9] D. Lei and B. Su, "A multi-class teaching–learning-based optimization for multi-objective distributed hybrid flow shop scheduling," *Knowl.-Based Syst.*, vol. 263, Mar. 2023, Art. no. 110252.

[10] W. Shao, Z. Shao, and D. Pi, "Modelling and optimization of distributed heterogeneous hybrid flow shop lot-streaming scheduling problem," *Exp. Syst. Appl.*, vol. 214, Mar. 2023, Art. no. 119151.

[11] V. Fernandez-Viagas, "A speed-up procedure for the hybrid flow shop scheduling problem," *Exp. Syst. Appl.*, vol. 187, Jan. 2022, Art. no. 115903.

[12] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan, "Optimization and approximation in deterministic sequencing and scheduling: A survey," *Ann. Math.*, vol. 5, pp. 287–326, Jan. 1979.

[13] B. Naderi, S. Gohari, and M. Yazdani, "Hybrid flexible flowshop problems: Models and solution methods," *Appl. Math. Model.*, vol. 38, pp. 5767–5780, Dec. 2014.

[14] D. L. Santos, J. L. Hunsucker, and D. E. Deal, "Global lower bounds for flow shops with multiple processors," *Eur. J. Oper. Res.*, vol. 80, no. 1, pp. 112–120, Jan. 1995.

[15] J. Carlier, "Scheduling jobs with release dates and tails on identical machines to minimize the makespan," *Eur. J. Oper. Res.*, vol. 29, no. 3, pp. 298–306, Jun. 1987.

[16] V. Fernandez-Viagas, P. Perez-Gonzalez, and J. M. Framinan, "Efficiency of the solution representations for the hybrid flow shop scheduling problem with makespan objective," *Comput. Operations Res.*, vol. 109, pp. 77–88, Sep. 2019.

[17] T. S. Arthanari and K. G. Ramamurthy, "An extension of two machines sequencing problem," *Opsearch*, vol. 8, no. 1, pp. 10–22, 1971.

[18] J. N. D. Gupta, "Two-stage, hybrid flowshop scheduling problem," *J. Oper. Res. Soc.*, vol. 39, no. 4, pp. 359–364, Apr. 1988.

[19] C. Sriskandarajah and S. P. Sethi, "Scheduling algorithms for flexible flowshops: Worst and average case performance," *Eur. J. Oper. Res.*, vol. 43, no. 2, pp. 143–160, Nov. 1989.

[20] J. N. D. Gupta and E. A. Tunc, "Schedules for a two-stage hybrid flowshop with parallel machines at the second stage," *Int. J. Prod. Res.*, vol. 29, no. 7, pp. 1489–1502, Jul. 1991.

[21] C.-Y. Lee and G. L. Vairaktarakis, "Minimizing makespan in hybrid flowshops," *Oper. Res. Lett.*, vol. 16, no. 3, pp. 149–158, Oct. 1994.

[22] F. Riane, A. Artiba, and S. E. Elmaghraby, "A hybrid three-stage flowshop problem: Efficient heuristics to minimize makespan," *Eur. J. Oper. Res.*, vol. 109, no. 2, pp. 321–329, Sep. 1998.

[23] H. Soewandi and S. E. Elmaghraby, "Sequencing three-stage flexible flowshops with identical machines to minimize makespan," *IIE Trans.*, vol. 33, no. 11, pp. 985–993, Nov. 2001.

[24] R. Ruiz and J. A. Vázquez-Rodríguez, "The hybrid flow shop scheduling problem," *Eur. J. Oper. Res.*, vol. 205, no. 1, pp. 1–18, 2010.

[25] S. A. Brah and L. L. Loo, "Heuristics for scheduling in a flow shop with multiple processors," *Eur. J. Oper. Res.*, vol. 113, no. 1, pp. 113–122, Feb. 1999.

[26] M.-C. Portmann, A. Vignier, D. Dardilhac, and D. Dezalay, "Branch and bound crossed with GA to solve hybrid flowshops," *Eur. J. Oper. Res.*, vol. 107, no. 2, pp. 389–400, Jun. 1998.

[27] J. Carlier and E. Neron, "An exact method for solving the multiprocessor flowshop," *RAIRO Oper. Res.*, vol. 34, no. 1, pp. 1–25, 2000.

[28] E. Néron, P. Baptiste, and J. N. D. Gupta, "Solving hybrid flow shop problem using energetic reasoning and global operations," *Omega*, vol. 29, no. 6, pp. 501–511, Dec. 2001.

[29] T. Kis and E. Pesch, "A review of exact solution methods for the non-preemptive multiprocessor flowshop problem," *Eur. J. Oper. Res.*, vol. 164, no. 3, pp. 592–608, 2005.

[30] S.-W. Lin, C.-Y. Cheng, P. Pourhejazy, K.-C. Ying, and C.-H. Lee, "New benchmark algorithm for hybrid flowshop scheduling with identical machines," *Exp. Syst. Appl.*, vol. 183, Nov. 2021, Art. no. 115422.

[31] C. D. Paternina-Arboleda, J. R. Montoya-Torres, M. J. Acero-Dominguez, and M. C. Herrera-Hernandez, "Scheduling jobs on a k-stage flexible flow-shop," *Ann. Operations Res.*, vol. 164, no. 1, pp. 29–40, Nov. 2008.

[32] D. Santos, J. Hunsucker, and D. Deal, "An evaluation of sequencing heuristics in flow shops with multiple processors," *Comput. Ind. Eng.*, vol. 30, pp. 681–692, Sep. 1996.

[33] V. Fernandez-Viagas, J. M. Molina-Pariente, and J. M. Framinan, "New efficient constructive heuristics for the hybrid flowshop to minimise makespan: A computational evaluation of heuristics," *Exp. Syst. Appl.*, vol. 114, pp. 345–356, Dec. 2018.

[34] E. Nowicki and C. Smutnicki, "The flow shop with parallel machines: A Tabu search approach," *Eur. J. Oper. Res.*, vol. 106, nos. 2–3, pp. 226–253, Apr. 1998.

[35] K. Alaykýran, O. Engin, and A. Döyen, "Using ant colony optimization to solve hybrid flow shop scheduling problems," *Int. J. Adv. Manuf. Technol.*, vol. 35, nos. 5–6, pp. 541–550, Nov. 2007.

[36] O. Engin and A. Döyen, "A new approach to solve hybrid flow shop scheduling problems by artificial immune system," *Future Gener. Comput. Syst.*, vol. 20, no. 6, pp. 1083–1095, 2004.

[37] Q.-K. Pan, L. Wang, J.-Q. Li, and J.-H. Duan, "A novel discrete artificial bee colony algorithm for the hybrid flowshop scheduling problem with makespan minimisation," *Omega*, vol. 45, pp. 42–56, Jun. 2014.

[38] C. Kahraman, O. Engin, İ. Kaya, and M. K. Yilmaz, "An application of effective genetic algorithms for solving hybrid flow shop scheduling problem," *Int. J. Comput. Intell. Syst.*, vol. 1, pp. 134–147, May 2008.

[39] D. Kizilay, M. F. Tasgetiren, Q.-K. Pan, and L. Wang, "An iterated greedy algorithm for the hybrid flowshop problem with makespan criterion," in *Proc. IEEE Symp. Comput. Intell. Prod. Logistics Syst. (CIPLS)*, Dec. 2014, pp. 16–23.

[40] R. Buddala and S. S. Mahapatra, "Improved teaching–learning-based and Jaya optimization algorithms for solving flexible flow shop scheduling problems," *J. Ind. Eng. Int.*, vol. 14, no. 3, pp. 555–570, Sep. 2018.

[41] S.-L. Jiang, C. Xu, L. Zhang, and Y. Ma, "A decomposition-based two stage online scheduling approach and its integrated system in the hybrid flow shop of steel industry," *Exp. Syst. Appl.*, vol. 213, Mar. 2023, Art. no. 119220.

[42] H. Qin, Y. Han, Y. Wang, Y. Liu, J. Li, and Q. Pan, "Intelligent optimization under blocking constraints: A novel iterated greedy algorithm for the hybrid flow shop group scheduling problem," *Knowl.-Based Syst.*, vol. 258, Dec. 2022, Art. no. 109962.

[43] Y. Liu, W. Shen, C. Zhang, and X. Sun, "Agent-based simulation and optimization of hybrid flow shop considering multi-skilled workers and fatigue factors," *Robot. Comput.-Integr. Manuf.*, vol. 80, Apr. 2023, Art. no. 102478.

[44] Q. N. H. Tran, N. Q. Nguyen, F. Yalaoui, L. Amodeo, and H. Chehade, "Improved formulations and new valid inequalities for a hybrid flow shop problem with time-varying resources and chaining time-lag," *Comput. Oper. Res.*, vol. 149, Jan. 2023, Art. no. 106018.

[45] J. Dong and C. Ye, "Green scheduling of distributed two-stage reentrant hybrid flow shop considering distributed energy resources and energy storage system," *Comput. Ind. Eng.*, vol. 169, Jul. 2022, Art. no. 108146.

[46] H. Gholami and H. Sun, "Toward automated algorithm configuration for distributed hybrid flow shop scheduling with multiprocessor tasks," *Knowl.-Based Syst.*, vol. 264, Mar. 2023, Art. no. 110309.
[47] Q.-K. Pan and Y. Dong, "An improved migrating birds optimisation for a hybrid flowshop scheduling with total flowtime minimisation," *Inf. Sci.*, vol. 277, pp. 643–655, Sep. 2014.
[48] M. Dell'Amico and S. Martello, "Optimal scheduling of tasks on identical parallel processors," *ORSA J. Comput.*, vol. 7, no. 2, pp. 191–200, May 1995.

YARONG CHEN received the B.S. and M.S. degrees in management science and engineering from Jiangsu University, Zhenjiang, Jiangsu, China, in 2003. Since 2003, she has been an Assistant Professor and an Associate Professor with the College of Mechanical and Electrical Engineering, Wenzhou University, Wenzhou, China. Her research interests include the simulation and optimization of complicated manufacturing systems, the integration optimization of production planning and scheduling, scheduling algorithms, and lean production.

YA-CHIH TSAI received the M.S. and Ph.D. degrees from the Department of Industrial Engineering and Management, Yuan Ze University. She is currently an Associate Professor with the Department of Hotel Management, Vanung University. Her current research interests include production scheduling and semiconductor manufacturing management.

FUH-DER CHOU received the M.S. degree in industrial engineering from Chung Yuan Christian University, in 1988, and the Ph.D. degree in industrial engineering and management from National Chiao Tung University, in 1997. He is currently a Professor with the College of Mechanical and Electronic Engineering, Wenzhou University, China. His research interests include production scheduling, semiconductor manufacturing management, and group technology.