Journal of Industrial and Systems Engineering
Vol. 14, No. 4, pp. 30-49
Autumn (November) 2022

New mathematical modelling and a constructive heuristic algorithm for integrated process planning and scheduling

Mohammad Reza Hosseinzadeh[1], Mehdi Heydari[1*], Mohammad Mahdavi Mazdeh[1]

[1] *Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran*

m_hosseinzadeh@ind.iust.ac.ir, mheydari@iust.ac.ir, mazdeh@iust.ac.ir

Abstract

Recent advances in manufacturing systems and multifunction machines have caused products to be produced through several alternative process plans. Therefore, the integration of process planning and scheduling, as two of the most critical functions, becomes essential to enhance manufacturing systems' productivity. Several different algorithms have solved the integrated process planning and scheduling (IPPS) problem in the literature. All proposed algorithms require a list of available process plans in advance (type-1). In this paper, an efficient mixed-integer linear programming (MILP) model is presented based on the term "combination." Besides, a type-2 priority-based heuristic algorithm (PBHA II) is proposed using dispatching rules with prioritizing jobs, combinations, and operations to solve the IPPS problems expressed by AND/OR graphs and with a makespan criterion. The MILP model and proposed heuristic algorithm are tested on the most challenging benchmark problems. Experimental results show the superiority of the MILP model over the best one in the literature, as well as the effectiveness and high performance of PBHA II. New upper bounds have been obtained in a short computational time for 7 of 24 most complex problems, which have been used by many researchers over the last two decades.

Keywords: Integrated process planning and scheduling, priority-based heuristic algorithm, dispatching rules

1-Introduction

Process planning and scheduling are the most important functions in a manufacturing system and significantly impact its flexibility and efficiency. Process planning determines the selection and sequence of production operations based on product design specifications as well as the required manufacturing resources, including machines, tools, and tool-approach directions (TADs). In general, a process plan identifies how a product can be manufactured according to engineering design. On the other hand, scheduling allocates limited manufacturing resources to the operation in the process plans over time, subject to the precedence relations in the process plan. These two functions are traditionally performed sequentially; Scheduling plans were generated after process plans had been determined. However, this method has some drawbacks (Li et al., 2010b, Li et al., 2010a). Unbalanced resource loads, unexpected bottlenecks, and the infeasibility of the generated process plan in the scheduling phase due to the unpredictable shop floor disturbances are the most important disadvantages. By integrating process planning and scheduling, the load of the resources is balanced, flow-time, work-in-process inventory, cycle time, and production costs are reduced (Lee and Kim, 2001).

---

*Corresponding author

Three kinds of flexibility can be considered in the integrated process planning and scheduling (IPPS) context: operation flexibility (OF), sequencing flexibility (SF), and processing flexibility (PF). OF refers to the possibility of performing an operation on different machines and is also called routing flexibility. Sequencing flexibility implies the availability of various permutations of manufacturing operations as long as they satisfy the precedence constraints to create a specific feature of a job, and processing flexibility means the possibility of producing the same manufacturing feature with an alternative set of operations. By taking all these flexibilities into account, although more reliable and stable plans are generated, the IPPS problem is much more complex.

From the representation point, the IPPS problems are either type-1 or type- 2. In type-1, all available process plans of each job are predetermined in advance as the input data for a solution algorithm in which one process plan should be selected for each job. In type-2, the precedence relations between the operations and all job flexibilities are expressed by the network graphs. In real-world manufacturing systems with high-level flexibilities, the type-2 representation is more practical. Moreover, type-1 IPPS problems can simply be converted to type-2. There is still no solution method based on type-2 representations to the best of our knowledge. This paper proposes an effective type -2 heuristic algorithm to solve the IPPS problems.

The remainder of the paper is organized as follows: In section 2, we review some relevant literature on integrating process planning and scheduling. Representation and the definition of the problem are presented in section 3. Section 4 introduces an efficient type-2 mixed-integer linear programming (MILP) model for the IPPS problem. The proposed constructive heuristic algorithm is elaborated in section 5. Section 6 evaluates the proposed mathematical model and solution method using the most challenging benchmark problems in the literature. The last section concludes the paper.

2-Literature review

Although mathematical model-based exact algorithms cannot achieve the optimal solution of real-world size IPPS problems in a reasonable computational time, mathematical modeling can accurately point out the characteristics of the problem and provide the basis for evaluating other algorithms. On the other hand, the ever-increasing improvement of computer capabilities enables researchers to apply mathematical approaches for even medium-size problems. In the IPPS mathematical modeling domain, the first attempt was made by Kim and Egbelu (1999). Tan and Khoshnevis (2004) proposed a polynomial mixed integer programming model (PMIPM) for the IPPS problem, but process and sequence flexibilities have not been considered in their model, nevertheless. By considering all types of flexibilities, a mathematical model was presented for the IPPS problem by Li et al. (2010a) with various objective functions based on sequence-based variables (Manne's approach (Manne, 1960)). Their model has been used in other research with some modifications (Li et al., 2012a, Li et al., 2012b, Luo et al., 2017). Özgüven et al. (2010) extended their model for flexible job shop scheduling problems by considering process plan flexibility. The model was evaluated by hypothetical test problems with different process plan and routing flexibility levels.

A more effective MILP model for the IPPS problem was proposed by Jin et al. (2015). They extended their model for the multi-objective case (Jin et al., 2016b). Three new MILP models for the IPPS problem, called Model-2, Model-3, and Model-4, were also developed by Jin et al. (2016a). Unlike previous MILP models in the literature, which assume that all the process plans are generated in advance (type-1), the proposed models are suitable for network graph-based instances (type-2). They can solve the instances expressed by an AND/OR graph. Their proposed models were established based on the term "combination," which is indispensable operations to complete the job without considering the precedence relationship between operations. Model-2 is based on position-based variables (Wagner's approach (Wagner, 1959)), while the other two models are constructed according to Manne's approach. Moreover, the common operations from all the combinations have been removed in Model-3. All the proposed MILP models have been tested on Kim's benchmark instances (Kim et al., 2003). Their experiments showed the superiority of Model-4, which can be regarded as the best IPPS mathematical model in the literature until now. As one of the most popular topics in scheduling, several papers have proposed various approaches for the IPPS problem over the past two decades. Due to the complexity of the problem, most of the proposed solution methods are based on heuristic search approaches. The most prominent published papers for the IPPS problem with the performance measure and assumptions considered in this research are summarized in table 1.

As it is clear, most proposed methods are metaheuristic-based algorithms, which are general-purpose and problem-independent algorithms that can be applied to almost any optimization problem. Regarding several constraints imposed on the IPPS problem, such as the precedence relationship between operations of a job and machine availability on the one hand and the random mechanism of metaheuristics for generating new solutions, on the other hand, metaheuristic-based algorithms require a feasibility procedure to repair the unfeasible solutions. Due to the large solution space of the IPPS problem, it may take a long time to search the entire problem space for feasible solutions randomly. Therefore, metaheuristic algorithms may not perform efficiently for the real-world size IPPS problem. To cope with these difficulties, heuristic-based algorithms as problem-dependent approaches are applied. Using the IPPS problem's particularities, heuristic approaches can find near-optimum or even global optimum solutions within a more reasonable time. However, a few heuristic methods can be found in the IPPS literature. A heuristic is presented by Bensmaine et al. (2014) for solving the IPPS problem in reconfigurable manufacturing systems using the two parameters they have defined, i.e., the availability time (AT) and selection index (SI). At each step, the operation with the highest SI is processed by the machine with minimum AT. Liu et al. (2020) applied two heuristic algorithms in two stages for the energy-efficient IPPS (EEIPPS) problem to minimize total tardiness and energy consumption. Dispatching rules, as the most popular constructive heuristics in scheduling, in conjunction with a priority-based assignment mechanism, are employed by Ausaf et al. (2015) to propose an efficient algorithm called the priority-based heuristic algorithm (PBHA) for the IPPS problem with makespan objective function. While introducing the concept of a chain as an independent subset of a job as well as the criticality of jobs and chains, jobs and chains are prioritized in this algorithm. Then operations are selected based on priorities while dispatching rules are incorporated to choose a processing machine for the selected operation.

Table 1. Solution methods proposed for the problem considered in this study

Study	**Solution Method**	**Study**	**Solution Method**
Zhu et al. (2022)	GH + GA	Uslu et al. (2022)	GA + ACO
Awad and Abd-Elaziz (2021)	modified GA	Wu and Li (2021)	HS
Barzanji et al. (2019)	logic-based benders decomposition	Li et al. (2019)	GA + VNS
Liu et al. (2018)	quantum-inspired hybrid algorithm	Keddari et al. (2018)	shifting bottleneck heuristic + TS + KA
Zhang and Wong (2016)	ACO	Zhang and Wong (2015)	object-coding GA
Jin et al. (2015)	hybrid honey bee mating optimization	Ausaf et al. (2015)	priority-based heuristic algorithm
Liu et al. (2016)	ACO	Wang et al. (2014)	improved ACO
Zhang and Wong (2014)	enhanced ACO	Li et al. (2012b)	active learning GA
Wong et al. (2012)	two-stage ACO	Lian et al. (2012)	imperialist competitive algorithm
Amin-Naseri and Afshari (2012)	hybrid GA	Lihong and Shengping (2012)	improved GA
Li et al. (2010a)	EA	Leung et al. (2010)	agent-based ACO
Li et al. (2010c)	GA + TS	Li et al. (2010d)	agent-based modified GA
Rajkumar et al. (2010)	GRASP	Hengyun et al. (2009)	PSO + SA
Shao et al. (2009)	modified GA	Guo et al. (2009)	PSO
Li et al. (2008)	GA	Tian et al. (2008)	immune algorithm
Chan et al. (2008)	GA with the dominant gene	Kim et al. (2007)	asymmetric multileveled SEA
Li and McMahon (2007)	SA	Park and Choi (2006)	GA
Zhao et al. (2006)	PSO + SA	Fuqing et al. (2006)	PSO + fuzzy inference system
Wong et al. (2006c)	online hybrid agent-based negotiation	Wong et al. (2006a)	hybrid-based agent negotiation
Wong et al. (2006b)	multi-agent negotiation	Kim et al. (2003)	SEA
Lee and Kim (2001)	simulation-based GA		

GH greedy heuristic, ***GA*** genetic algorithm, ***ACO*** ant colony optimization, ***HS*** harmony search, ***VNS*** variable neighborhood search, ***TS*** tabu search, ***KA*** kangaroo algorithm, ***EA*** evolutionary algorithm, ***GRASP*** greedy randomized adaptive search procedure, ***PSO*** particle swarm optimization, ***SA*** simulated annealing, ***SEA*** symbiotic evolutionary algorithm

However, all solution algorithms mentioned above assume that all available process plans are determined in advance. That is, these approaches are type-1. Nevertheless, it is sometimes time-consuming to identify and list all the process plans for jobs, especially where multiple process plans are shown by AND/OR graphs. To fulfil such gaps in the literature, a simple and efficient type-2 constructive priority-based heuristic algorithm (PBHA II) is proposed in this paper based on the effective type-2 mathematical modelling presented in section 4.

3-Problem definition

The IPPS problem is defined as *n* jobs to be accomplished by *M* machines. Each job contains a number of operations, each of which is processed by one of the alternative machines with known processing time. A set of alternative process plans are available for each job. The goal is to assign a process plan for each job and a machine for each operation as well as find the best sequence of processing the operations of the jobs considering the precedence constraints among operations in order to corresponding objectives can be achieved. The makespan minimization is selected as the objective here.

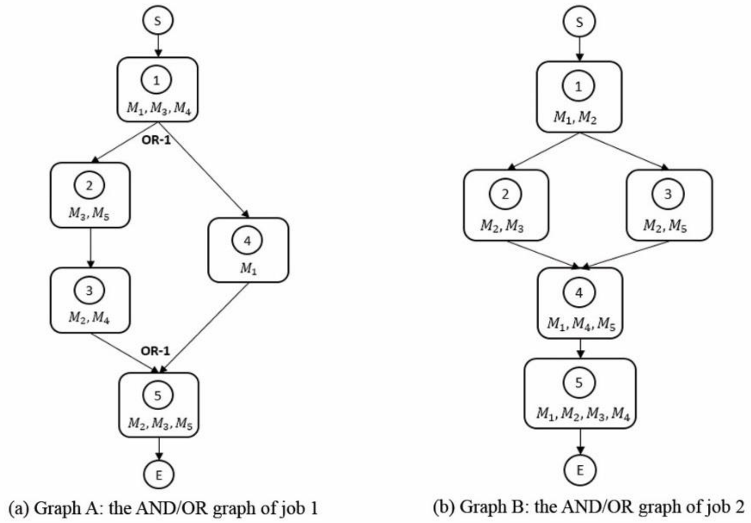

Fig 1. An IPPS problem instance with two jobs and five machines

The alternative process plans and precedence constraints are usually specified by disjunctive AND/OR graphs by which three types of flexibilities in process planning can be described. OR marks in the graph illustrate the operation combinations of the related job (process flexibility) because the operations on only one OR link-path should be processed. An IPPS problem, adopted by Zhang and Wong (2014), is shown in figure 1 with two jobs and two corresponding AND/OR graphs that should be processed by five available machines. Graph A and graph B describe the process plans of job 1 and job 2, respectively. According to the graphs, two process plans for each job can be recognized as follows:

$$T_1 = \begin{cases} O_1 \rightarrow O_2 \rightarrow O_3 \rightarrow O_5 \\ O_1 \rightarrow O_4 \rightarrow O_5 \end{cases}, \qquad T_2 = \begin{cases} O_1 \rightarrow O_2 \rightarrow O_3 \rightarrow O_4 \rightarrow O_5 \\ O_1 \rightarrow O_3 \rightarrow O_2 \rightarrow O_4 \rightarrow O_5 \end{cases}$$

T_1 and T_2 are the set of process plans of job 1 and job 2, respectively. Since graph A has one OR mark, job 1 can be performed through two operation combinations, while job 2 has only one operation combination, as demonstrated below:

$$H_1 = \{O_1, O_2, O_3, O_5\}, \{O_1, O_4, O_5\}, \qquad H_2 = \{O_1, O_2, O_3, O_4, O_5\}$$

H_1 and H_2 represent the set of operation combinations of job 1 and job 2, respectively. In other words, job 1 can be done by $\{O_1, O_2, O_3, O_5\}$ or $\{O_1, O_4, O_5\}$ while all operations of graph B are needed to perform job 2. In this example, both jobs have operation flexibility, while only job 1 has process flexibility, and sequence flexibility can only be observed in job 2. Furthermore, the following assumptions are considered:

- A machine can execute only one operation at a given time.
- Different operations from one job cannot be processed at the same time
- All processing times are deterministic, known in advance.
- All jobs and machines are independent and available at time zero.
- Job pre-emption is not taken into account.
- Once a job's operation is finished, it will be immediately transferred to another machine (the transport time is negligible).
- Setup times are negligible or included in the processing times.

4-Mathematical modelling

In this section, model-4 of Jin et al. (2016a), as the best mathematical model in the literature till now, is modified to present an efficient type-2 model for the IPPS problem. A dummy job with one process plan (and one operation combination) is considered in the proposed model. The dummy job has one operation on each machine with zero processing time. A job that is processed immediately after the dummy job, indeed, is the first scheduled job on the related machine. The notations, parameters, and decision variables used to explain the model are as follows:

4-1-Subscripts, notations, and sets

i, i'	jobs,
j, j'	operations,
h, h'	combinations,
k	machines,
N	the set of jobs,
M	the set of machines,
H_i	the set of operation combinations for job i,
$NO_{i,h}$	the set of operations of h-th combination of job i,
$O_{i,j}$	the j-th operation of job i,
$R_{i,j}$	the set of available machines for $O_{i,j}$.

4-2-Parameters

$t_{i,j,k}$	the processing time of $O_{i,j}$ on machine k,
$n = \|N\|$	the number of jobs,
$m = \|M\|$	the number of machines,
$V_{i,j,j'}$	1, if the $O_{i,j}$ is an immediate predecessor of $O_{i,j'}$ according to the network graph of job i; 0, otherwise,
$Q_{i,j,j'}$	1, if the $O_{i,j}$ should be processed before $O_{i,j'}$ directly or indirectly according to the network graph of job i; 0, otherwise,
A	a very large positive number.

4-3-Variables

C_{max}	makespan,
$X_{i,h}$	1, if the h-th alternative combination of operations is selected to accomplish job i; 0, otherwise,

$Z_{i,j,h,k}$	1, if the h-th combination is selected for job i and $O_{i,j}$ is processed on machine k; 0, otherwise,
$Y_{i,j,j'}$	1, if $O_{i,j}$ precedes the operation $O_{i,j'}$; 0, otherwise,
$U_{i,j,i',j'}$	1, if the j'-th operation of job i' is processed after the j-th operation of job i on machine k; 0, otherwise,
$C_{i,j}$	the completion time of the j-th operation of job i.

With regard to the above definitions, the modified type-2 model of the IPPS is formulated as follows:

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

subject to:

$$\sum_{h\in H_i} X_{i,h} = 1, \ \ \forall i \in N \tag{2}$$

$$\sum_{k\in R_{i,j}} Z_{i,j,h,k} + \left(1 - X_{i,h}\right) = 1, \ \ \forall i \in N, h \in H_i, j \in NO_{i,h} \tag{3}$$

$$C_{i,j} \geq \sum_{k\in R_{i,j}} t_{i,j,k} Z_{i,j,h,k}, \ \ \forall i \in N, h \in H_i, j \in NO_{i,h} \tag{4}$$

$$C_{i,j'} \geq C_{i,j} + \sum_{k\in R_{i,j'}} t_{i,j',k}\, Z_{i,j',h,k} - A\left(1 - X_{i,h}\right), \forall i \in N, h \in H_i, j, j' \in NO_{i,h}, j \neq j', V_{i,j,j'} = 1 \tag{5}$$

$$Y_{i,j,j'} + Y_{i,j',j} = 1, \ \ \forall i \in N - \{0\}, h \in H_i, j, j' \in NO_{i,h}, j < j', Q_{i,j,j'} + Q_{i,j',j} = 0 \tag{6}$$

$$C_{i,j'} \geq C_{i,j} + \sum_{k\in R_{i,j'}} t_{i,j',k}\, Z_{i,j',h,k} - A\left(2 - Y_{i,j,j'} - X_{i,h}\right), \quad \forall i \in N - \{0\}, h \in H_i, j, j' \in NO_{i,h}, j \neq j', Q_{i,j,j'} + Q_{i,j',j} = 0 \tag{7}$$

$$C_{i',j'} \geq C_{i,j} + t_{i',j',k} - A\left(3 - U_{i,j,i',j'} - Z_{i,j,h,k} - Z_{i',j',h',k}\right), \quad \forall i, i' \in N, i \neq 0, i < i', h \in H_i, h' \in H_{i'}, j \in NO_{i,h}, j' \in NO_{i',h'}, k \in R_{i,j} \cap R_{i',j'} \tag{8}$$

$$C_{i,j} \geq C_{i',j'} + t_{i,j,k} - A\left(2 + U_{i,j,i',j'} - Z_{i,j,h,k} - Z_{i',j',h',k}\right), \quad \forall i, i' \in N, i \neq 0, i < i', h \in H_i, h' \in H_{i'}, j \in NO_{i,h}, j' \in NO_{i',h'}, k \in R_{i,j} \cap R_{i',j'} \tag{9}$$

$$Ci,j \ \ i_{i,h_{max}} \tag{10}$$

$$C_{i,j} \geq 0, \tag{11}$$

$$X_{i,h}, Z_{i,j,h,k}, Y_{i,j,j'}, U_{i,j,i',j'} \in \{0,1\} \tag{12}$$

As presented by equation (1), the objective is to minimize the maximum completion time of jobs (makespan). Constraint set (2) ensures that only one combination of operations is selected for each job. Constraint set (3) is incorporated into the model to make sure that each operation is assigned to only one machine. Constraint set (4) ensures that the completion time of only selected operations can take on positive values. Constraint set (5) is incorporated into the model to compute the completion time of two operations of a job that have immediate precedence in the related network graph. Constraint sets (6) and (7) determine the sequence of the operations that have no explicit precedence relationship with each other. Constraint sets (8) and (9) are included in the model to schedule different operations on the same machine. Constraint set (10) is used to calculate the makespan. Constraint sets (11) and (12) define the decision variables.

Modifications of the presented MILP model compared to Model-4 include the use of two-dimensional variables $C_{i,j}$ for the completion time of operations instead of three-dimensional variables $C_{i,h,j}$, the replacement of the inefficient constraint set (35) of Model-4 with a new constraint set (4), and innovative changes to constraint sets (5-9) to make the model more efficient.

To speed up the convergence, the only existing lower bound in the literature, i.e., the maximum of the shortest combination of operations, presented by Lihong and Shengping (2012), is formulated as follows:

$$Cmax_{i>0}\left(\min_{h\in H_i}\left(\sum_{j\in NO_{i,h}}\sum_{k\in R_{i,j}} t_{i,j,k}Z_{i,j,h,k}\right)\right)_{max} \tag{13}$$

Equation set (13) is converted to its linear form using a binary variable $B_{i,h}$, and as a result, the following sets of constraints are added to the basic model:

$$C \sum_{j\in NO_{i,h}}\sum_{k\in R_{i,j}} t_{i,j,k}Z_{i,j,h,k}\left(1-B_{i,h}\right)\ \ \{0\}_i \quad {}_{max} \tag{14}$$

$$\sum_{h\in H_i} B_{i,h} \geq 1,\ \ \forall i \in N \tag{15}$$

$$B_{i,h} \leq X_{i,h},\ \ \forall i \in N \tag{16}$$

$$B_{i,h} \in \{0,1\} \tag{17}$$

$B_{i,h}$determines which operation combination of job i generates the shortest path. Constraint set (15) ensures that at least one combination gives the shortest path for each job. If a combination of operations is not selected, it is not taken into account to determine the lower bound. Constraint set (16) is incorporated into the model for this reason. We called the resulting model the enhanced model.

5-Constructive heuristic algorithm

As mentioned in section 2, all solution algorithms presented for the IPPS problem assume all available process plans are known and listed in advance. In real-world problems, it is sometimes difficult and time-consuming to identify and generate all the process plans for a job according to its AND/OR graph, especially for jobs with high sequence flexibility. For instance, too many process plans can be identified for a job with the AND/OR graph in figure 2. It should be noted that the depicted graph is relatively simple, and it has no OR junction. However, there is only one combination of operations through which the job can perform.

In this section, according to the term "combination," the type-2 version of the priority-based heuristic algorithm (PBHA) proposed by Ausaf et al. (2015) is developed. The resulting constructive heuristic algorithm is called PBHA II. Furthermore, priority functions and the strategy of operation selection are totally improved. PBHA II is not only applicable to the IPPS problems represented by complex graphs, but also it obtains significant results.

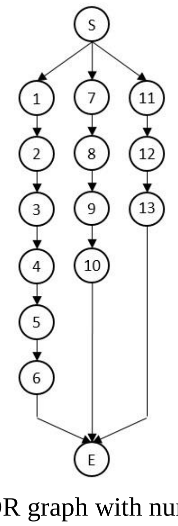

Fig 2. An AND/OR graph with numerous process plans

5-1-Job selection mechanism

Inspired by the lower bound of IPPS problems, the minimum processing time (JT_i) required for each job is obtained as follows:

$$JT_i = \min_h \left(\sum_{j \in NO_{i,h}} t^*_{i,j} \right) \tag{18}$$

Where,

$$t^*_{i,j} = \begin{cases} t_{i,j,k_0}, & \text{if machine } k_0 \text{ is selected for operation } O_{i,j} \\ \min_k t_{i,j,k}, & \text{if no machine is selected for } O_{i,j} \text{ yet} \end{cases} \tag{19}$$

The maximum JT_i $\left(JT_c = \max_i JT_i\right)$ is called critical time, and the job with the processing time equal to $JT_c$is defined as a critical job. To ensure that the highest priority is assigned to the critical job, the job score (JS) is calculated using the following relationship:

$$JS_i = JT_i - \min_i JT_i + 1 \tag{20}$$

Finally, the priority of each job is calculated as a probability (JP) as follows:

$$JP_i = \frac{JS_i}{\sum_{i \in N} JS_i} \tag{21}$$

5-2-Combination selection mechanism

An approach similar to job selection is applied to select a combination of operations for each job. Firstly, for each job, the minimum processing time of each combination ($T_{i,h}$) is determined by equation (22). Clearly, the combinations with less $T_{i,h}$ should have more priority for selection. It is also reasonable that a job with more priority has more chances to perform through its shortest path. Therefore, equations (23) and (24) are used to calculate the combination score (CS):

$$T_{i,h} = \sum_{j \in NO_{i,h}} \min_k t_{i,j,k} \tag{22}$$

$$CS_{i,h} = \max_h T_{i,h} - T_{i,h} + D_{i,h} \tag{23}$$

$$D_{i,h} = \begin{cases} JP_i * \dfrac{\sum_h T_{i,h}}{|H_i|} + 1, & \text{if } T_{i,h} = \min_h T_{i,h} \\ 1, & \text{otherwise} \end{cases} \tag{24}$$

Furthermore, $|H_i|$ is the number of combinations of job i. Then, the probability which indicates the priority of combination h of job *i* ($CP_{i,h}$) is computed as follows:

$$CP_{i,h} = \frac{CS_{i,h}}{\sum_{h \in H_i} CS_{i,h}} \tag{25}$$

5-3-Operation selection mechanism

As a prioritization approach for entering operations into the schedule, the ranked positional weight (RPW) method of assembly line balancing is adopted. The weight of operation $O_{i,j}$ where combination

h_0 is selected (W_{i,j,h_0}) is obtained as follows:

$$W_{i,j,h_0} = \min_k t_{i,j,k} + \sum_{\substack{j' \in NO_{i,h_0}, \\ Q_{i,j,j'}=1}} \min_k t_{i,j',k} \quad \forall j \in NO_{i,h_0} \tag{26}$$

At first, executable operations are determined. An operation is executable if all its immediate precedence operations are scheduled before, or if it does not have any predecessor. Afterward, executable operations are ranked according to their weight in descending order, and the first operation is selected to enter the schedule.

5-4-Machine selection mechanism

Dispatching rules are used to select a processing machine for the selected operation. A large number of dispatching rules have been applied in the scheduling domain. However, with regard to makespan as our optimization criterion, initial experiments show that two popular dispatching rules are more effective than others for this problem: 1) shortest processing time (SPT) and 2) earliest starting time (EST). SPT selects the machine with the shortest processing time for the operations, while EST selects the machine so that the earliest starting time for the operations will result. Since the best result of PBHA II is obtained using SPT and EST, other dispatching rules are ignored due to the computational time consideration.

5-5-Dispatching rules-based population classification

In this part of the algorithm, the population-based procedure proposed by Ausaf et al. (2015) is utilized for the selected dispatching rules. In each iteration, a population of solutions is generated and divided into two groups. One dispatching rule is used for each group to choose processing machines. At the end of each iteration, the population portion of dispatching rules is changed based on the average makespan for each group. The number of individuals in the worse group is reduced by *r*, and the number of individuals in the better group is increased by *r* instead. In this way, PBHA II uses the better dispatching rule more for the next iteration according to the prior iteration results.

5-6-PBHA II stages

The flow chart of PBHA II is depicted in figure 3. The algorithm is initiated by calculating the scores and then the probabilities of all jobs using equations (18-21). Afterward, the scores and probabilities of combinations are obtained for each job using equations (22-25). From here, the iterations of the algorithm are started. Due to PBHA II taking advantage of population-based approaches, N_{pop} individuals are first generated and divided into two groups equally. Steps are repeated for each individual.

The computed probabilities of combinations are used to select one combination of operations for each job. For the executable operations of the selected combination of a job, the weight is calculated using equation (26), and the operation with maximum weight is selected. Depending on which group individuals belong, the processing machines are determined by SPT or EST, and the operation is scheduled. Once a machine is selected for an operation, the scores and probabilities of the jobs are updated.

These steps are repeated until the operations of all jobs are scheduled. At the end of each iteration, N_{pop} solutions are generated, and the best one is sorted. The best solution of all iterations is presented as the algorithm output. Moreover, the average makespan of the generated solution is used to alter the population portion of each group for the next iteration. PBHA II is executed for a certain number of iterations unless the given IPPS problem's lower bound is achieved.

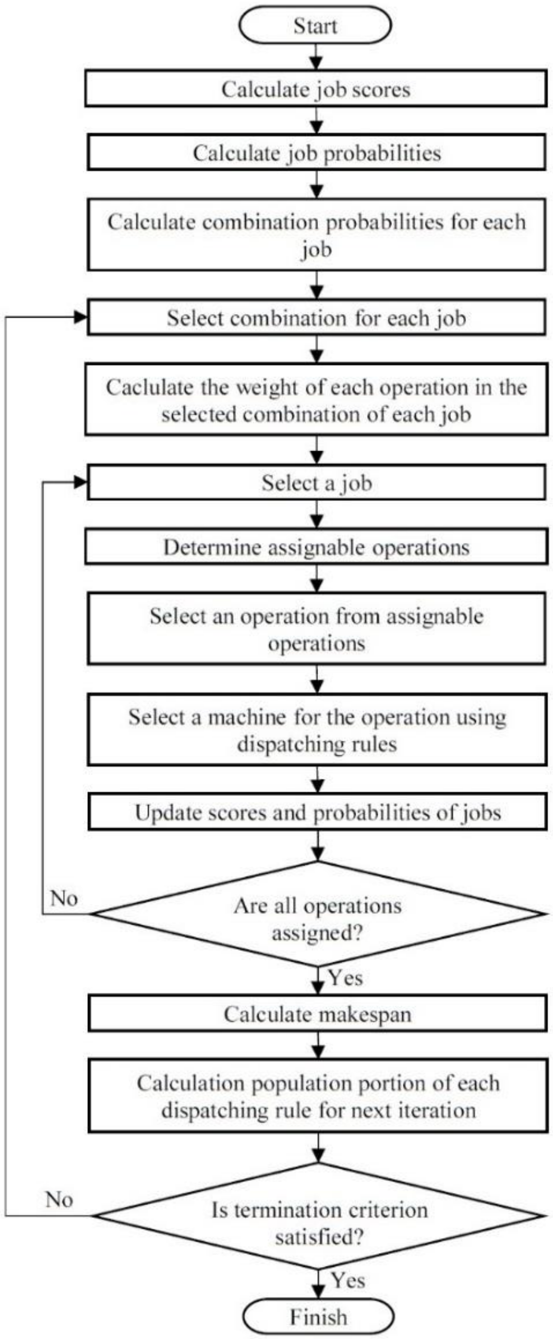

Fig 3. Flow chart for PBHA II

6-Computational results

Both the proposed mathematical model and heuristic algorithm are evaluated in this section. The mathematical model is executed using Cplex solver (version 12.8) within the GAMS (version 25.0.2) environment, and PBHA II is coded in Matlab 9.4. All experiments are implemented on a computer with an Intel Core i5-3470@3.20 GHz with 8 GB RAM.

6-1-Model evaluation

The proposed type-2 model is compared with Model-4 of Jin et al. as the best mathematical model in the literature using Kim's benchmark (Kim et al., 2003). This benchmark includes 24 test-bed instances, constructed with 18 jobs with various kinds and levels of flexibilities and 15 machines. The total number of operations of the problems varies from 79 to 300. Also, the 18 jobs can be done through up to 12 combinations of operations. A time limitation of 3600 s is imposed. Table 2 presents the number of single equations, single variables, and discrete variables for the proposed basic model, the proposed enhanced model, and Model-4 of Jin et al. separately.

Table 2. Comparison of three models in terms of the number of equations and variables

No.	TNO*	Model-4			Basic model		
		Single equations	Single variables	Discrete variables	Single equations	Single variables	Discrete variables
1	79	21,030	4,582	4,464	5,879	2,084	1,988
2	100	127,135	7,692	7,402	30,368	3,348	3,231
3	121	629,357	11,675	11,000	172,658	5,443	5,305
4	95	164,166	7,322	6,990	38,777	2,938	2,826
5	96	138,425	7,156	6,800	38,336	3,172	3,059
6	109	212059	9233	8838	59,512	4,646	4,520
7	99	127,406	6,131	5,745	34,354	2,701	2,585
8	96	170,237	7,014	6,639	40,461	3,099	2,986
9	105	219,087	11,050	10,728	63,394	5,198	5,076
10	132	99,113	11,949	11,695	25,433	5,184	5,035
11	168	1,013,833	19,847	19,019	265,249	8,695	8,510
12	146	370,023	15,078	14,594	95,387	6,082	5,919
13	154	469,251	16,506	15,908	120,443	7,651	7,480
14	151	449,206	13,181	12,536	110,325	5,487	5,319
15	149	391,128	18,861	18,424	106,945	8,475	8,309
16	179	263,561	21,063	20,656	62,166	8,684	8,488
17	221	1,426,076	32,116	31,152	367,676	13,775	13,537
18	191	688,827	24,101	23,414	170,699	9,711	9,503
19	205	794,980	27,408	26,658	213,696	12,028	11,806
20	195	673,811	20,850	20,090	161,310	8,409	8,197
21	201	887,824	30,763	30,067	229,510	13,372	13,154
22	256	1,271,020	42,745	41,849	332,927	18,135	17,862
23	256	1,578,490	40,862	39,845	402,360	17,018	16,745
24	300	1,823,562	55,655	54,574	463,092	22,954	22,637

* Total Number of Operations

It can be inferred from Table 2 that our basic model has more than 70% fewer constraints and also, more than 55% fewer variables than Model-4. For example, for Problem 24, the number of discrete variables of the proposed model is less than half of Model-4. Comparing the two models shows the significant efficiency of the proposed model than previous models. Since the number of constraints and variable augmentation in the enhanced model is not considerable, it is expected to be faster than the basic model because of faster convergence.

The results of the three discussed models with corresponding computational times and the gap reported by the solver are listed in Table 3. Gap% indicates the quality of a solution. It is calculated as (BF-BP)/BF, where BF and BP are the best-found value and the best possible value of the objective function, respectively. Obviously, a solution with a gap of 0% is the optimum solution. The lower bound (LB) is achieved for 14 of 24 problems using the basic proposed model. Furthermore, the enhanced model can find the optimum solution for 16 problems in a given time. In comparison, Model-4 cannot obtain even a feasible solution for 10 problems where there are more than 9 jobs.

Table 3. Computational results of Kim's benchmark

No.	Model-4			Basic model			Enhanced model			LB
	C_{max}	Gap (%)	Time (s)	C_{max}	Gap (%)	Time (s)	C_{max}	Gap (%)	Time (s)	
1	427	46.10	3600	427[a)]	32.32	3600	427[b)]	0	1.67	427
2	346	42.80	3600	343[a)]	28.86	3600	343[b)]	0	52.22	343
3	408	82.40	3600	346	35.26	3600	344[b)]	0	321.23	344
4	313	26.30	3600	306[a)]	0	577	306[b)]	0	51.23	306
5	326	39.30	3600	318[a)]	28.3	3600	318[b)]	0	94.05	318
6	438	63.70	3600	427[a)]	36.3	3600	427[b)]	0	165.16	427
7	373	34.60	3600	372[a)]	31.45	3600	372[b)]	0	26.58	372
8	346	51.40	3600	343[a)]	35.57	3600	343[b)]	0	143.8	343
9	433	54.30	3600	427[a)]	34.19	3600	427[b)]	0	33.84	427
10	445	42.70	3600	427[a)]	40.28	3600	427[b)]	0	911.41	427
11	-	-	3600	347	29.68	3600	345	0.29	3600	344
12	406	45.70	3600	318[a)]	19.81	3600	318[b)]	0	3567	318
13	684	76.80	3600	427[a)]	50.59	3600	427[b)]	0	415.23	427
14	469	49.80	3600	372[a)]	31.45	3600	372[b)]	0	1018.14	372
15	456	56.60	3600	427[a)]	42.15	3600	427[b)]	0	287.31	427
16	-	-	3600	427[a)]	40.28	3600	427[b)]	0	382.05	427
17	-	-	3600	401	39.15	3600	390	12.05	3600	344
18	-	-	3600	332	23.19	3600	327	2.75	3600	318
19	-	-	3600	442	49.77	3600	439	2.73	3600	427
20	-	-	3600	390	34.62	3600	382	2.62	3600	372
21	-	-	3600	430	59.77	3600	427[b)]	0	2516.67	427
22	-	-	3600	528	58.9	3600	517	17.41	3600	427
23	-	-	3600	509	49.9	3600	471	21.02	3600	372
24	-	-	3600	630	59.52	3600	534	20.04	3600	427

- a feasible solution is not available after 3600 s.
a) LB achieved.
b) an optimum solution

6-2-Algorithm evaluation

Various benchmark problems have been generated in previous research to illustrate the performance of the methods proposed for IPPS problems. However, there is neither process nor sequence flexibility in most cases. In this situation, the benchmark problems were presented for the flexible job shop problem (FJSP), which can be regarded as the IPPS problem without multiple routings. Also, in some other test problems, flexibilities and complexity are low, so they cannot compare the corresponding algorithms' performance.

Two more challenging benchmark problems are used to evaluate the performance of PBHA II. Experiment 1 is one of the most used FJSP instances with no process and sequence flexibilities, while experiment 2 contains the problems with all three kinds of flexibility on different levels. The parameters of PBHA II for all runs are set: N_{pop}=20, maximum iterations=50, and r=2.

Experiment 1

The data of this experiment is presented by Chryssolouris et al. (1992), constructed with 10 jobs, 9 machines, and a total number of operations of 35. It has been applied by Jain and Elmaraghy (1997), Wong et al. (2006c), Lihong and Shengping (2012), and Ausaf et al. (2015). Using PBHA II, an improved makespan of 5102 is achieved. Table 4 shows the results of the solution methods. Figure 4 presents the Gantt chart for the problem obtained by PBHA II.

Table 4. Experimental results of experiment 1

Solution Methods	GA (Jain and Elmaraghy, 1997)	oHAN (Wong et al., 2006c)	IGA (Lihong and Shengping, 2012)	PBHA (Ausaf et al., 2015)	PBHA II
Makespan	6456	6574	5998	5388	5102

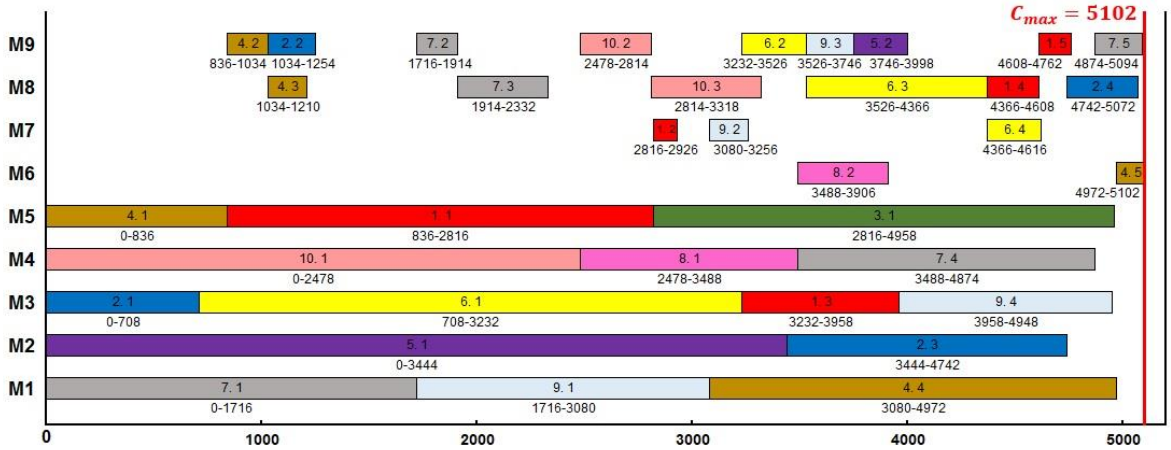

Fig 4. The Gantt chart for experiment 1

Experiment 2

For the second experiment, the benchmark problems used for model evaluation are applied again to evaluate PBHA II performance. Due to the different complexity and flexibility levels of 24 test-bed instances, it has been a popular basis for assessing solution methods in the IPPS domain. PBHA II is run for each test problem 10 times, and the best and average makespan are recorded.

The results are compared with the most significant state-of-the-art algorithms used for the experiment: the symbiotic evolutionary algorithm (SEA) (Kim et al., 2003), the hybrid algorithm (HA) (Li et al., 2010c), the improved genetic algorithm (IGA) (Lihong and Shengping, 2012), the imperialist competitive algorithm (ICA) (Lian et al., 2012), the active learning genetic algorithm (ALGA) (Li et al., 2012b), the enhanced ant colony optimization heuristic (E-ACO) (Zhang and Wong, 2014), the object-coding genetic algorithm (OCGA) (Zhang and Wong, 2015), honey bee mating optimization algorithm (HBMO) (Jin et al., 2015), the priority-based heuristic algorithm (PBHA) (Ausaf et al., 2015), and ant colony optimization algorithm (ACO) (Zhang and Wong, 2016).

The best makespan of each test problem obtained by PBHA II and the comparison with other algorithms are listed in table 5. Table 6 indicates that the algorithms presented the best results. As shown in Table 6, PBHA II can reach the lower bound (LB) for 18 problems, i.e., optimum solutions for these problems are found by PBHA II. Also, improved results are obtained for 7 problems, including problem 11, in which its optimum solutions are yielded. To sum up, PBHA II can improve the results or reach their lower bounds for all problems. The Gantt charts for problems 11, 17, and 24 are presented in figures 5-7. It should be noted that makespan values reported for IGA are less than the corresponding lower bounds for some test problems (No. 5, 8, and 12). In these cases, the reported results are ignored.

The average makespan and the required CPU time are presented in Table 7 and Table 8, respectively. Since the corresponding values of HA, ICA, and ALGA are unavailable, these algorithms are omitted from Table 6. As it reveals, PBHA II achieved improvements in 10 out of 24 problems, while the lower bound was achieved for all runs of the other 14 problems. In terms of CPU time, similar to PBHA, since the proposed heuristic algorithm is designed based on the inherent characteristics of the IPPS problem, PBHA II requires significantly less computational time than other metaheuristic algorithms. Even compared to PBHA, our algorithm needs less time because fewer dispatching rules are used to select processing machines or different specifications of the computers on which the algorithms are run. However, the CPU time of small-size and less complex problems is slightly longer than HBMO. The corresponding results for large-scale and more complex instances are significantly less than other metaheuristic algorithms.

However, the main advantage of PBHA II as a type-2 algorithm in terms of the solution time is that contrary to type-1 methods, it requires no list of process plans prior to initiating. It is sometimes considerably time-consuming to determine the list of all available process plans while it is plainly depicted by an AND/OR graph. For example, as a simple job, job 11 in the final experiment has several process plans that should be generated initially. This preparation phase is not included in PBHA II.

Table 5. Comparison of algorithms according to the best-achieved makespan

No.	SEA	HA	IGA	ICA	ALGA	E-ACO	OCGA	HBMO	PBHA	ACO	PBHA II
1	428	427	427	427	427	427	427	427	427	427	427
2	343	343	343	343	343	343	343	343	343	343	343
3	347	345	344	345	344	344	344	345	344	344	344
4	306	306	306	306	306	306	306	306	306	307	306
5	319	322	-	319	321	318	318	319	318	318	318
6	438	429	427	435	427	427	427	427	427	427	427
7	372	372	372	372	372	372	372	372	372	372	372
8	343	343	-	343	347	343	343	343	343	343	343
9	428	427	427	427	427	427	427	427	427	427	427
10	443	430	427	440	427	427	427	427	427	427	427
11	369	369	368	367	369	348	348	347	347	364	344*
12	328	327	-	327	327	322	318	326	318	332	318
13	452	436	429	457	436	427	427	427	427	427	427
14	381	380	386	390	380	373	372	372	376	382	372
15	434	427	427	432	427	427	427	427	427	427	427
16	454	446	433	466	446	429	427	427	427	438	427
17	431	423	415	443	423	377	370	377	394	398	360*
18	379	377	364	384	377	357	351	326	352	378	323*
19	490	476	450	490	474	431	427	427	445	451	427
20	447	432	429	440	438	386	384	377	426	412	375*
21	477	446	433	466	447	428	427	427	427	430	427
22	534	518	491	529	513	444	446	432	475	480	431*
23	498	470	465	495	470	413	394	391	455	453	390*
24	587	544	532	577	548	460	458	441	526	525	440*

* An improved result.

Table 6. The best algorithms, according to the best-achieved makespan

No.	LB	Best
1	427	HA, IGA, ICA, ALGA, E-ACO, OCGA, HBMO, PBHA, ACO, PBHA II*
2	343	SEA, HA, IGA, ICA, ALGA, E-ACO, OCGA, HBMO, PBHA, ACO, PBHA II*
3	344	IGA, ALGA, E-ACO, OCGA, PBHA, ACO, PBHA II*
4	306	SEA, HA, IGA, ICA, ALGA, E-ACO, OCGA, HBMO, PBHA, PBHA II*
5	318	E-ACO, OCGA, PBHA, ACO, PBHA II*
6	427	IGA, ALGA, E-ACO, OCGA, HBMO, PBHA, ACO, PBHA II*
7	372	SEA, HA, IGA, ICA, ALGA, E-ACO, OCGA, HBMO, PBHA, ACO, PBHA II*
8	343	SEA, HA, ICA, E-ACO, OCGA, HBMO, PBHA, ACO, PBHA II*
9	427	HA, IGA, ICA, ALGA, E-ACO, OCGA, HBMO, PBHA, ACO, PBHA II*
10	427	IGA, ALGA, E-ACO, OCGA, HBMO, PBHA, ACO, PBHA II*
11	344	PBHA II*
12	318	OCGA, PBHA, PBHA II*
13	427	E-ACO, OCGA, HBMO, PBHA, ACO, PBHA II*
14	372	OCGA, HBMO, PBHA II*
15	427	HA, IGA, ALGA, E-ACO, OCGA, HBMO, PBHA, ACO, PBHA II*
16	427	OCGA, HBMO, PBHA, PBHA II*
17	344	PBHA II
18	318	PBHA II
19	427	OCGA, HBMO, PBHA II*
20	372	PBHA II
21	427	OCGA, HBMO, PBHA, PBHA II*
22	427	PBHA II
23	372	PBHA II
24	427	PBHA II

* LB achieved.

Table 7. Comparison of average makespan values

No.	Average Makespan							
	SEA	IGA	E-ACO	OCGA	HBMO	PBHA	ACO	PBHA II
1	437.6	427	427.1	427	427	427	427.1	427
2	349.7	344.5	343.1	343.5	345.5	343	343.2	343
3	355.2	351	345	346.4	346.8	344	347.1	344
4	306.2	307.4	307.6	310.1	307.6	306	309.4	306
5	323.7	-	319.6	323	323.3	318	320.4	318
6	443.8	427	427.1	427	427	427	427.5	427
7	372.4	372.7	372	373.3	372	372	372	372
8	348.3	357	343.3	343.5	344.7	343	344.4	343
9	434.9	427	427.1	427	427	427	427	427
10	456.5	431.6	427.6	427.1	427	427	429.2	427
11	378.9	379.7	350.2	350.6	348.3	354.8	368.3	345.7
12	332.8	323.7	323.4	324.7	326	318	337.7	318
13	469	442.8	427.6	427.2	427	428.3	433.2	427
14	402.4	415.3	374.3	377.4	372.3	384.8	385.1	372
15	445.2	427.4	427.3	427	427	427	427.3	427
16	478.8	449.4	430.9	428.1	427.3	442.1	442.2	427
17	448.9	426	381.2	383.1	384.4	408.1	414.3	367.4
18	389.6	373.6	361.5	354.5	332.9	358.1	383.7	327.9
19	508.1	471.3	434.9	433.7	430.7	459	460.4	429.7
20	453.8	446.6	392.4	390.5	391.4	433.8	421.6	378.8
21	483.2	447.8	429.4	427	427.6	427	435.7	427
22	548.3	508.1	447.2	452.9	439.7	490.6	484.2	438.7
23	507.5	477.8	420.3	410	403.8	468.6	462.8	394.1
24	602.2	548.5	479.3	471.2	455.7	548	531.1	451.4

Table 8. Comparison of CPU times

No.	CPU time (sec)							
	SEA	IGA	E-ACO	OCGA	HBMO	PBHA	ACO	PBHA II
1	60.5	11	17	4.5	0.1	1.77	4.1	0.53
2	68.9	11	15	6.5	0.7	2.11	4	0.92
3	81.7	11	14	7.9	1.4	2.02	4.9	1.02
4	65.6	8	14	4.4	0.3	2.09	3	0.83
5	63.5	8	11	6	0.9	1.75	3	0.58
6	73.3	13	20	7.4	0.4	2.28	7	0.75
7	69	9	11	4.1	0.3	1.88	3	0.42
8	67.3	17	13	6.2	1.3	1.86	4	0.51
9	73.2	9	21	5.7	0.1	1.89	6	0.38
10	136	17	34	10.9	0.6	3.09	10	1.28
11	165.8	16	31	12.2	1.9	3.16	9.4	1.43
12	143.4	13	24	8.7	1.6	2.7	6.8	1.59
13	161.2	19	39	15.3	4.7	3.42	13	1.12
14	150.8	16	26	11.2	3.9	3.06	8	1.65
15	156	14	33	10.7	0.5	3.06	12	0.9
16	333.6	23	50	27.8	5.6	4.17	16.5	1.16
17	435.2	23	64	27.5	337.3	4.34	18.7	1.74
18	357	20	53	26.4	35.4	4.01	15.3	2.08
19	417.8	28	78	30.5	17.5	4.58	21	1.97
20	384	26	55	25.9	464.2	4.28	15.1	2.13
21	392.4	24	67	26.5	3	4.38	20.1	1.81
22	1033.3	27	121	33.5	53.7	6.06	30.1	2.73
23	1016.6	26	93	31.5	259.7	5.96	26	2.28
24	1622.7	39	186	48.5	343	7.45	40	3.17

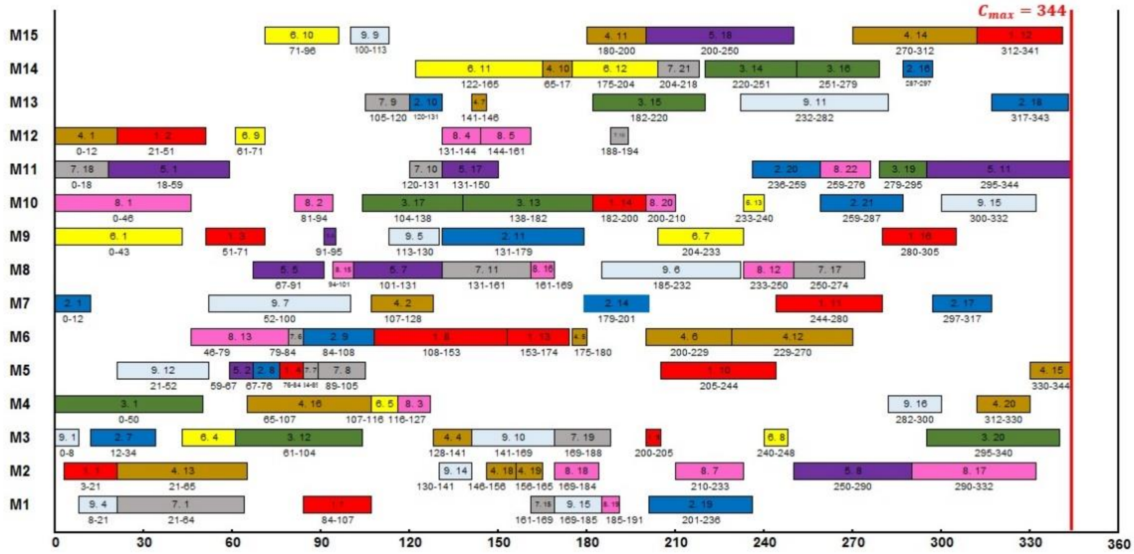

Fig 5. The Gantt chart for experiment 2, problem 11

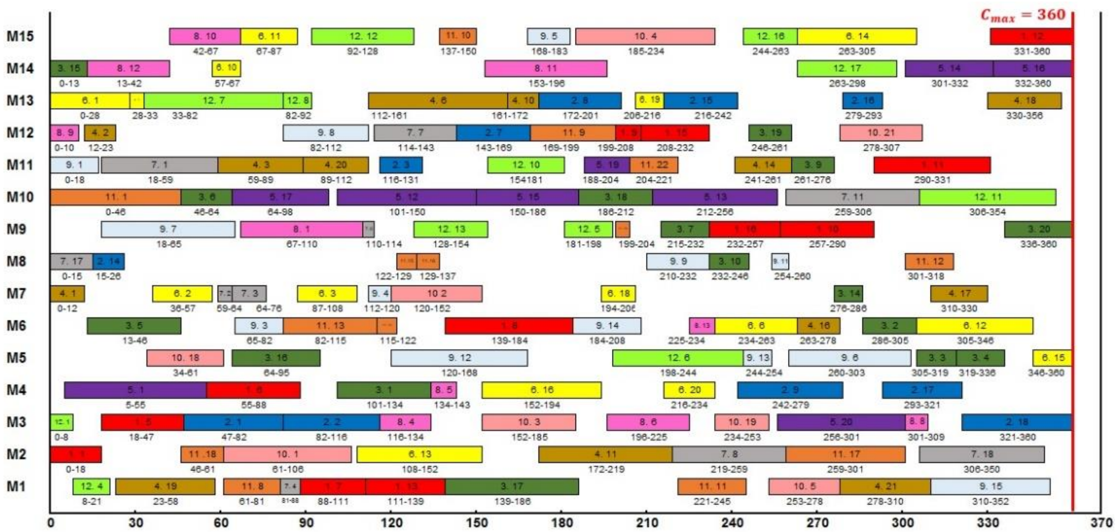

Fig 6. The Gantt chart for experiment 2, problem 17

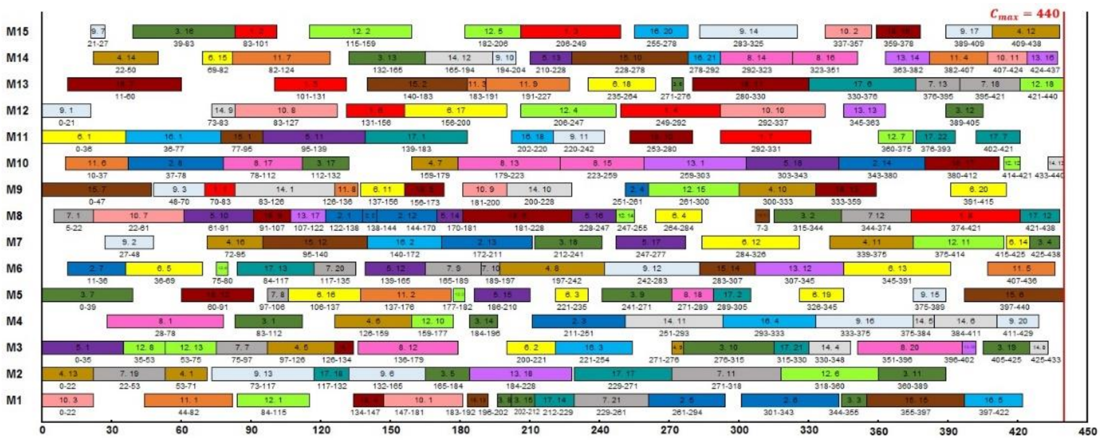

Fig 7. The Gantt chart for the last problem of experiment 2

7-Conclusions

Considering type-2 IPPS problems, an efficient combination-based MILP model has been presented. The lower bound of the IPPS problems has been linearized and incorporated into the model to speed up the convergence when the solver executes it. Furthermore, a type-2 constructive heuristic algorithm, named type-2 priority-based heuristic algorithm (PBHA II), has been presented to solve the IPPS problem. According to the defined scores for jobs and combinations, this algorithm utilizes priority assignment mechanisms. It is based on the weights calculated by the ranked positional weight (RPW) method for operations.

Most complex benchmark problems have been selected to evaluate the proposed model and algorithm. Experimental results indicate the efficiency and effectiveness of the MILP model compared to those of the state-of-the-art mathematical models in the literature. Moreover, encouraging results have been obtained for PBHA II. It can achieve either optimal solutions or improved results for all experiments under consideration. Due to the simplicity of the proposed method, the outstanding results are yielded in less computational time than other methods. Further research directions involve the consideration of sequence-dependent setup times. PBHA II also can be extended to cope with a multi-objective IPPS problem.

References

Amin-Naseri, M. R. & Afshari, A. J. (2012). A hybrid genetic algorithm for integrated process planning and scheduling problem with precedence constraints. *The International Journal of Advanced Manufacturing Technology,* 59**,** 273-287.

Ausaf, M. F., Gao, L., Li, X. & Al Aqel, G. (2015). A priority-based heuristic algorithm (PBHA) for optimizing integrated process planning and scheduling problem. *Cogent Engineering,* 2**,** 1070494.

Awad, M. A. & Abd-Elaziz, H. M. An Efficient Modified Genetic Algorithm for Integrated Process Planning-Job Scheduling. (2021) International Mobile, Intelligent, and Ubiquitous Computing Conference, MIUCC 2021, 2021. 319-323.

Barzanji, R., Naderi, B. & Begen, M. A. (2019). Decomposition algorithms for the integrated process planning and scheduling problem. *Omega***,** 102025.

Bensmaine, A., Dahane, M. & Benyoucef, L. (2014). A new heuristic for integrated process planning and scheduling in reconfigurable manufacturing systems. *International Journal of Production Research,* 52**,** 3583-3594.

Chan, F. T. S., Chung, S. H. & Chan, L. Y. (2008). An introduction of dominant genes in genetic algorithm for FMS. *International Journal of Production Research,* 46**,** 4369-4389.

Chryssolouris, G., Pierce, J. E. & Dicke, K. (1992). A decision-making approach to the operation of flexible manufacturing systems. *International Journal of Flexible Manufacturing Systems,* 4**,** 309-330.

Fuqing, Z., Aihong, Z., Dongmei, Y. & Yahong, Y. A Hybrid Particle Swarm Optimization(PSO) Algorithm Schemes for Integrated Process Planning and Production Scheduling. (2006) 6th World Congress on Intelligent Control and Automation, 21-23 June 2006 2006. 6772-6776.

Guo, Y. W., Li, W. D., Mileham, A. R. & Owen, G. W. (2009). Optimisation of integrated process planning and scheduling using a particle swarm optimisation approach. *International Journal of Production Research,* 47**,** 3775-3796.

Hengyun, Z., Wenhua, Y. & Guangxia, B. A particle swarm optimization for integrated process planning and scheduling. (2009) IEEE 10th International Conference on Computer-Aided Industrial Design & Conceptual Design, 26-29 Nov. 2009 2009. 1070-1074.

Jain, A. K. & Elmaraghy, H. A. (1997). Production scheduling/rescheduling in flexible manufacturing. *International Journal of Production Research,* 35**,** 281-309.

Jin, L., Tang, Q., Zhang, C., Shao, X. & Tian, G. (2016a). More MILP models for integrated process planning and scheduling. *International Journal of Production Research,* 54**,** 4387-4402.

Jin, L., Zhang, C. & Shao, X. (2015). An effective hybrid honey bee mating optimization algorithm for integrated process planning and scheduling problems. *The International Journal of Advanced Manufacturing Technology,* 80**,** 1253-1264.

Jin, L., Zhang, C., Shao, X., Yang, X. & Tian, G. (2016b). A multi-objective memetic algorithm for integrated process planning and scheduling. *The International Journal of Advanced Manufacturing Technology,* 85**,** 1513-1528.

Keddari, N., Mebarki, N., Shahzad, A., & Sari, Z. (2018). Solving an integration process planning and scheduling in a flexible job shop using a hybrid approach. In *Computational Intelligence and Its Applications: 6th IFIP TC 5 International Conference, CIIA 2018, Oran, Algeria, May 8-10, 2018, Proceedings 6* (pp. 387-398). Springer International Publishing.

MOHAMED, O. & DJEBBAR, B., eds. Computational Intelligence and Its Applications, (2018) Cham. Springer International Publishing, 387-398.

Kim, K. H. & Egbelu, P. J. (1999). Scheduling in a production environment with multiple process plans per job. *International Journal of Production Research,* 37**,** 2725-2753.

Kim, Y. K., Kim, J. Y. & Shin, K. S. (2007). An asymmetric multileveled symbiotic evolutionary algorithm for integrated FMS scheduling. *Journal of Intelligent Manufacturing,* 18**,** 631-645.

Kim, Y. K., Park, K. & Ko, J. (2003). A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. *Computers & Operations Research,* 30**,** 1151-1171.

Lee, H. & Kim, S.-S. (2001). Integration of Process Planning and Scheduling Using Simulation Based Genetic Algorithms. *The International Journal of Advanced Manufacturing Technology,* 18**,** 586-590.

Leung, C., Wong, T., Mak, K.-L. & Fung, R. Y. (2010). Integrated process planning and scheduling by an agent-based ant colony optimization. *Computers & Industrial Engineering,* 59**,** 166-180.

Li, W. & Mcmahon, C. A. (2007). A simulated annealing-based optimization approach for integrated process planning and scheduling. *International Journal of Computer Integrated Manufacturing,* 20**,** 80-95.

Li, X., Gao, L. & Li, W. (2012a). Application of game theory based hybrid algorithm for multi-objective integrated process planning and scheduling. *Expert Systems with Applications,* 39**,** 288-297.

Li, X., Gao, L., Pan, Q., Wan, L. & Chao, K.-M. (2019). An Effective Hybrid Genetic Algorithm and Variable Neighborhood Search for Integrated Process Planning and Scheduling in a Packaging Machine Workshop. *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* 49**,** 1933-1945.

Li, X., Gao, L. & Shao, X. (2012b). An active learning genetic algorithm for integrated process planning and scheduling. *Expert Systems with Applications,* 39**,** 6683-6691.

Li, X., Gao, L., Shao, X., Zhang, C. & Wang, C. (2010a). Mathematical modeling and evolutionary algorithm-based approach for integrated process planning and scheduling. *Computers & Operations Research,* 37**,** 656-667.

Li, X., Gao, L., Zhang, C. & Shao, X. (2010b). A review on integrated process planning and scheduling. *International Journal of Manufacturing Research,* 5**,** 161-180.

Li, X., Gao, L., Zhang, G., Zhang, C., & Shao, X. (2008). A genetic algorithm for integration of process planning and scheduling problem. In *Intelligent Robotics and Applications: First International Conference, ICIRA 2008 Wuhan, China, October 15-17, 2008 Proceedings, Part II 1* (pp. 495-502). Springer Berlin Heidelberg.

Li, X., Shao, X., Gao, L. & Qian, W. (2010c). An effective hybrid algorithm for integrated process planning and scheduling. *International Journal of Production Economics,* 126**,** 289-298.

Li, X., Zhang, C., Gao, L., Li, W. & Shao, X. (2010d). An agent-based approach for integrated process planning and scheduling. *Expert Syst. Appl.,* 37**,** 1256-1264.

Lian, K., Zhang, C., Gao, L. & Li, X. (2012). Integrated process planning and scheduling using an imperialist competitive algorithm. *International Journal of Production Research,* 50**,** 4326-4343.

Lihong, Q. & Shengping, L. (2012). An improved genetic algorithm for integrated process planning and scheduling. *The International Journal of Advanced Manufacturing Technology,* 58**,** 727-740.

Liu, M., Yi, S. & Wen, P. (2018). Quantum-inspired hybrid algorithm for integrated process planning and scheduling. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* 232**,** 1105-1122.

Liu, N., Zhang, Y. F. & Lu, W. F. (2020). Energy-efficient integration of process planning and scheduling in discrete parts manufacturing with a heuristic-based two-stage approach. *The International Journal of Advanced Manufacturing Technology,* 106**,** 2415-2432.

Liu, X., Ni, Z. & Qiu, X. (2016). Application of ant colony optimization algorithm in integrated process planning and scheduling. *The International Journal of Advanced Manufacturing Technology,* 84**,** 393-404.

Luo, G., Wen, X., Li, H., Ming, W. & Xie, G. (2017). An effective multi-objective genetic algorithm based on immune principle and external archive for multi-objective integrated process planning and scheduling. *The International Journal of Advanced Manufacturing Technology,* 91**,** 3145-3158.

Manne, A. S. (1960). On the Job-Shop Scheduling Problem. *Operations Research,* 8**,** 219-223.

Özgüven, C., Özbakır, L. & Yavuz, Y. (2010). Mathematical models for job-shop scheduling problems with routing and process plan flexibility. *Applied Mathematical Modelling,* 34**,** 1539-1548.

Park, B. J. & Choi, H. R. A Genetic Algorithm for Integration of Process Planning and Scheduling in a Job Shop. (2006) Berlin, Heidelberg. Springer Berlin Heidelberg, 647-657.

Rajkumar, M., Asokan, P., Page, T. & Arunachalam, S. (2010). A GRASP algorithm for the Integration of Process Planning and Scheduling in a flexible job-shop. *International Journal of Manufacturing Research,* 5**,** 230-251.

Shao, X., Li, X., Gao, L. & Zhang, C. (2009). Integration of process planning and scheduling—a modified genetic algorithm-based approach. *Computers & Operations Research,* 36**,** 2082-2096.

Tan, W. & Khoshnevis, B. (2004). A linearized polynomial mixed integer programming model for the integration of process planning and scheduling. *Journal of Intelligent Manufacturing,* 15**,** 593-605.

Tian, Y., Jiang, P. & Zheng, M. (2008). An Immune Algorithm for the integration of process planning and scheduling. *International Journal of Materials and Product Technology,* 33**,** 95.

Uslu, M. F., Uslu, S. & Bulut, F. (2022). An adaptive hybrid approach: Combining genetic algorithm and ant colony optimization for integrated process planning and scheduling. *Applied Computing and Informatics,* 18**,** 101-112.

Wagner, H. M. (1959). An integer linear-programming model for machine scheduling. *Naval Research Logistics Quarterly,* 6**,** 131-140.

Wang, J., Fan, X., Zhang, C. & Wan, S. (2014). A graph-based ant colony optimization approach for integrated process planning and scheduling. *Chinese Journal of Chemical Engineering,* 22**,** 748-753.

Wong, T., Leung, C., Mak, K.-L. & Fung, R. Y. (2006a). Dynamic shopfloor scheduling in multi-agent manufacturing systems. *Expert Systems with Applications,* 31**,** 486-494.

Wong, T., Zhang, S., Wang, G. & Zhang, L. (2012). Integrated process planning and scheduling–multi-agent system with two-stage ant colony optimisation algorithm. *International Journal of Production Research,* 50**,** 6188-6201.

Wong, T. N., Leung, C. W., Mak, K. L. & Fung, R. Y. K. (2006b). An agent-based negotiation approach to integrate process planning and scheduling. *International Journal of Production Research,* 44**,** 1331-1351.

Wong, T. N., Leung, C. W., Mak, K. L. & Fung, R. Y. K. (2006c). Integrated process planning and scheduling/rescheduling—an agent-based approach. *International Journal of Production Research,* 44**,** 3627-3655.

Wu, X. & Li, J. (2021). Two layered approaches integrating harmony search with genetic algorithm for the integrated process planning and scheduling problem. *Computers and Industrial Engineering,* 155.

Zhang, L. & Wong, T. N. (2015). An object-coding genetic algorithm for integrated process planning and scheduling. *European Journal of Operational Research,* 244**,** 434-444.

Zhang, L. & Wong, T. N. (2016). Solving integrated process planning and scheduling problem with constructive meta-heuristics. *Information Sciences,* 340–341**,** 1-16.

Zhang, S. & Wong, T. N. (2014). Integrated process planning and scheduling: an enhanced ant colony optimization heuristic with parameter tuning. *Journal of Intelligent Manufacturing,* 29**,** 585-601.

Zhao, F., Zhu, A., Ren, Z., & Yang, Y. (2006, June). Integration of process planning and production scheduling based on a hybrid PSO and SA algorithm. In *2006 International Conference on Mechatronics and Automation* (pp. 2290-2295). IEEE.

Zhu, X., Guo, X., Wang, W. & Wu, J. (2022). A Genetic Programming-Based Iterative Approach for the Integrated Process Planning and Scheduling Problem. *IEEE Transactions on Automation Science and Engineering,* 19**,** 2566-2580.