Computers & Industrial Engineering 168 (2022) 108135

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Non-identical parallel machines batch processing problem with release dates, due dates and variable maintenance activity to minimize total tardiness

Pedram Beldar[a,e], Milad Moghtader[b,f], Adriana Giret[c,*], Amir Hossein Ansaripoor[d]

[a] Department of Business Development, Sepid Makian Company, Rasht, Guilan, Iran
[b] Vista Samaneh Asa Company, Tehran, Iran
[c] VRAIN - Valencian Research Institute for Artificial Intelligence, Universitat Politècnica de València, Valencia, Spain
[d] School of Management, Curtin Business School, Kent Street, Bentley, WA 6102, Australia
[e] Department of Industrial Engineering, Payame Noor University, Tehran, Iran
[f] Department of Computer Engineering, Islamic Azad University, Rasht, Guilan, Iran

ARTICLE INFO

Keywords:
Minimization of total tardiness
Parallel machines
Batch processing machines
Scheduling with maintenance activity

ABSTRACT

Combination of job scheduling and maintenance activity has been widely investigated in the literature. We consider a non-identical parallel machines batch processing (BP) problem with release dates, due dates and variable maintenance activity to minimize total tardiness. An original mixed integer linear programming (MILP) model is formulated to provide an optimal solution. As the problem under investigation is known to be strongly NP-hard, two meta-heuristic approaches based on Simulated Annealing (SA) and Variable Neighborhood Search (VNS) are developed. A constructive heuristic method (H) is proposed to generate initial feasible solutions for the SA and VNS. In order to evaluate the results of the proposed solution approaches, a set of instances were randomly generated. Moreover, we compare the performance of our proposed approaches against four meta-heuristic algorithms adopted from the literature. The obtained results indicate that the proposed solution methods have a competitive behaviour and they outperform the other meta-heuristics in most instances. Although in all cases, H + SA is the most performing algorithm compared to the others.

1. Introduction

The batch processing (BP) machine scheduling problem and the machine scheduling problem with associated maintenance operations are two challenging problems in the literature of machine scheduling. In BP, a machine/processor is capable of dealing with a set of jobs simultaneously. In fact, the idea of BP is to group jobs into batches on every single machine and to schedule the formed batches. In real manufacturing systems, machines/processors may become unavailable due to maintenance operations, the need for repairs, sudden failures, and etc. However, many research studies in the machine scheduling literature assume that machines/processors are always ready for use during the manufacturing process. Hence, in this research, so as to have a more practical machine scheduling problem, we consider the minimization of total tardiness for an unrelated parallel machines BP problem under the constraints of release dates and flexible/variable maintenance activity. The main aim of such a research domain is to adopt hands-on and useful approaches for scheduling jobs when production halts as a result of maintenance issues.

This study was originally motivated by a food industry application, originating in Lahijan, Gilan province, Iran, where the cookies called Koloocheh are produced on a large scale. Cookies (jobs) are shaped by machines with many batter nozzles, and they are baked in ovens at a fixed temperature for a specified period of time. The number of trays of cookies that can be baked simultaneously (i.e., a batch) is determined by the capacity of the oven. In order to ensure that all cookies are well baked and there are no half-baked cookies, the processing time of a batch is defined by the longest processing time among all cookies in that batch. A cookie can be kept in the oven longer than its pre-set baking time, but not removed from the oven before the pre-set baking time. Once the baking of a batch has begun, it cannot be interrupted. No cookies can be added or removed from a batch in the oven until the

---

[*] Corresponding author.
E-mail address: agiret@dsic.upv.es (A. Giret).

https://doi.org/10.1016/j.cie.2022.108135
Received 17 August 2021; Received in revised form 21 February 2022; Accepted 27 March 2022

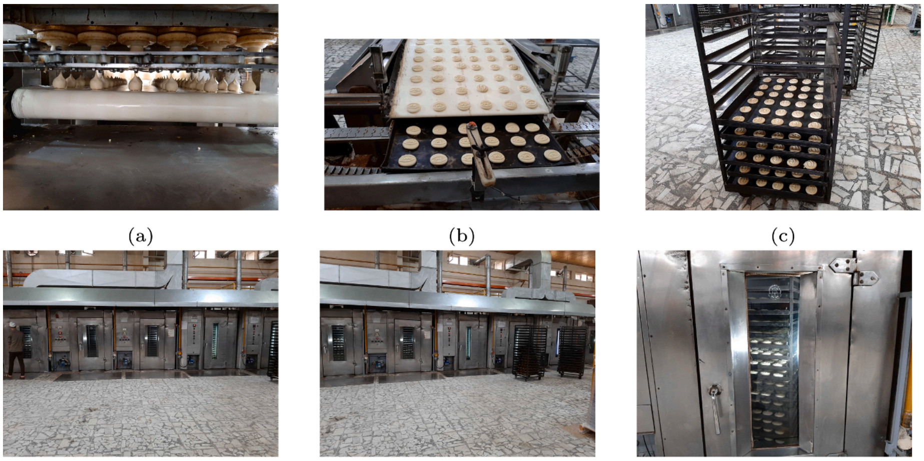

(a) (b) (c)
(d) (e) (f)

Fig. 1. Production Process of cookies called Koloocheh.

whole batch has been baked.The objective is to bake all cookies in the minimum time. Although cookies may differ in shape, their weight and thickness must be almost identical. However, since nozzles are not usually well-maintained, their performance worsens gradually with time. Hence, in reality, the weight and the thickness of cookies are not consistent, and each cookie tray may require a different baking time (processing time). Fig. 1 illustrates the production process used for these types of cookies. The maintenance task must be performed in a pre-planned time window corresponding to the time when the maintenance staff are available. In this research, the maintenance operation involves a cleaning operation which the length increases in accordance with its starting time. The oven-cleaning process consists of two activities. Firstly, there is the removal of dust particles entering the oven when its door is opened or closed by a particular device. Secondly, while the cookies are being baked, an amount of steam and smoke associated with the water and oil in the cookie batter sticks to the inner wall of the oven, compromising the quality and increasing the product’s vulnerability to microbial contamination. Therefore, in order to guarantee the safety and the quality of the cookies, the ovens must be cleaned within the pre-planned time frame. Lack of effectual scheduling of these maintenance operations for BP machines (ovens) can increase the possibility that the products will be unsafe and of poor quality.

Table 1
Single machine with flexible/variable machine maintenance.

Authors	Objective function	Model	Methodology
Yang et al. (2002)	C_{max}		Heuristic
Chen (2006a)	$\sum T_j$	✓	CPLEX
Chen (2006b)	$\overline{F}$	✓	CPLEX
Chen (2008)	C_{max}	✓	CPLEX, Heuristic
Sbihi and Varnier (2008)	T_{max}		Branch-and-bound (B&B), Heuristic
Low et al. (2010)	C_{max}		Heuristic
Yang and Yang (2010)	C_{max}	✓	polynomial time solution algorithm
Xu and Yin (2011)	C_{max}		polynomial time solution algorithm
Yang et al. (2011)	$\sum C_j$		Heuristic, Dynamic programming, B&B
Luo et al. (2015)	$C_{max}, \sum C_j, L_{max}, \sum U_j$		Polynomial time optimal algorithm
Luo and Ji (2015)	$C_{max}, \sum C_j$		Polynomial time approximation algorithm
Ying et al. (2016)	$\overline{L}, T_{max}, \sum F_j$ and $\overline{T}$		Exact algorithm
Cui and Lu (2017)	C_{max}		Heuristic, B&B
Wang et al. (2017)	$\overline{L}, T_{max}, \sum F_j$ and $\overline{T}$	✓	Exact algorithms
Wang et al. (2018)	C_{max}	✓	Branch-and-price
Wang et al. (2019)	adjustment time and idle time of machine	✓	Heuristic
Detti et al. (2019)	$C_{max}, \sum C_j$	✓	Heuristic
Xu et al. (2020)	$C_{max}, \sum F_j$	✓	Heuristics or exact-solution approaches

In this research study, at first, a mathematical formulation is constructed to optimally solve the problem using small-sized instances. Then, we propose an effective heuristic method in which three interconnected decisions are made regarding: how to allocate jobs into the batches, how to sequence the formed batches on different machines, and how to arrange maintenance activities on each machine. Two meta-heuristic approaches based on Simulated Annealing (SA) and Variable Neighborhood Search (VNS) are also developed to solve the problem at hand. Both search techniques take advantage of the initial solution returned by the proposed heuristic technique (H) to accelerate the search process. In addition, the proposed techniques have been evaluated and compared with several recent meta-heuristics to analyze the behaviour in different instances. These meta-heuristics are a hybrid of Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) proposed by Beldar and Costa (2018), a Biased Random-Key GA and Differential Evolution (BRKGA-DE) proposed by Kong, Liu, Pei, Cheng, and Pardalos (2020), a hybrid of Artificial Bee Colony (ABC) and Tabu Search (TS) proposed by Lu, Liu, and Pei (2018), and an Iterated Greedy (IG) algorithm proposed by Arroyo, Leung, and Tavares (2019). Due to the high efficiency of PSO, GA, DE, ABC and TS in solving challenging scheduling problems (see e.g., Hulett, Damodaran, & Amouie (2017); Zhou, Pang, Chen, & Chou (2018a); Ding, Schulz, Shen, Buscher, & Lü (2021); Huang, Wang, & Jiang (2020); Li, Meng, Liang, & Zhao (2015); Zhou, Xie, Du, & Pang (2018b); Zhou, Liu, Chen, & Li (2016);Zhou et al. (2021);Arroyo & Leung (2017);Shahvari & Logendran (2017);Al-Salamah (2015)), we are motivated to employ the aforementioned algorithms for the comparison as the most relevant ones to the problem at hand.

The rest of the paper is organized as follows. In Section 2, we provide the research background, giving an overview of the most relevant research studies of the problem under investigation. In Section 3, we describe the problem we are trying to solve. In Section 4, we discuss our proposed mathematical formulation for the problem. The search techniques, including both heuristic and meta-heuristics are discussed in Section 5. The generation of the test instances and computational results are explained in Section 6. Section 7 concludes the paper with a recapitulation of the main points and suggestions for future research.

2. Research Background

Combination of job scheduling and maintenance activity has been widely investigated in the literature so far. There are two main research streams that focus on this combination: job scheduling with fixed maintenance and job scheduling with flexible/variable maintenance. In the first category, both beginning time and the duration of the maintenance activity are pre-set. A great number of research studies have focused their efforts on this kind of classification in different machine environments, such as single machine ones (see e.g. the recent articles by Zammori, Braglia, & Castellano (2014);Liu, Wang, Chu, & Chu (2016);Sun & Geng (2019)), parallel machines (see e.g. Ruiz-Torres, Paletta, & M'Hallah (2017);Avalos-Rosales, Angel-Bello, Álvarez, & Cardona-Valdés (2018); Zhang, Liu, Lin, & Wu (2020)), to name a few. In the second category, the beginning time of the maintenance task is a decision variable that is set by the decision-maker, and the duration of the maintenance is a positive and increasing function of its commencement time. In this research, we focus on flexible maintenance activity. Therefore, we first provide an overview of variable maintenance activity in a single machine environment; then, we examine the related research works on parallel machine environments.

2.1. *Single machine scheduling with flexible/variable maintenance activity*

Given that this research focuses on an unrelated parallel machine scheduling problem, we summarize several related works that deal with single machine problems along with flexible/variable machine maintenance. More precisely, Table 1 classifies the most important literature contributions, with respect to the objective functions, mathematical model, and the solution approaches for a single machine environment.

Despite the mathematical formulations and the solution approaches reported in the relevant literature presented in Table 1 are highly useful and effective for single machine environments, they are unsuitable for the parallel machines environment.

2.2. *Scheduling of parallel machines with flexible/variable maintenance activity*

In the following, we present in chronological order the main articles on parallel machines combined with flexible/variable maintenance activity.

Xu, Yin, and Li (2010) address identical parallel machines with flexible maintenance activity to minimize makespan. The time lapse between any two successive maintenance tasks is set at a predetermined interval. The time required to do one maintenance task on a processor/machine is an ascending function of the total processing time of the jobs that are dealt with after its last maintenance. They develop an approximation algorithm to solve the problem.

Wang and Wei (2011) consider an identical parallel machines problem with machine maintenance in which the duration of the maintenance task is contingent upon its beginning time. Two separate objective functions are taken into consideration: the total absolute differences in completion times and the total absolute differences in waiting times. They demonstrate that the problems are polynomially solvable.

Cheng, Hsu, and Yang (2011) study an unrelated parallel machines problem in combination with maintenance activity to minimize the total completion time or the total machine load. At most one maintenance task is executed on every single machine at any time during the planning horizon. The duration of the maintenance task increases linearly with its beginning time. They demonstrate that the problems are optimally solvable in polynomial time.

Hsu, Ji, Guo, and Yang (2013) address unrelated parallel machines problems in which the maintenance duration is a linear function of its beginning time. Throughout the planning horizon, there should be at most one maintenance task performed on each machine. The objectives is to minimize the total completion time and the total machine load. They show that the problems are optimally solvable in polynomial time.

Alfares, Mohammed, and Ghaleb (2021) consider the minimization of the makespan on a two-machine job scheduling problem with aging effects and maintenance operations. They assume that the number and the positions of maintenance stops are variable. Integer linear programming formulations are constructed for both the problem with maintenance and without maintenance in order to solve the problem in smaller sizes. They also propose six heuristic approaches to solve the large-sized problems.

The studies above do not include approaches for BP machines, which are the focus of our research. Hence, in the next subsection, we closely examine BP machines in terms of maintenance activity.

2.3. BP machines with maintenance activity

Few research works have considered BP machines with maintenance activity so far. Zarook, Rezaeian, Tavakkoli-Moghaddam, Mahdavi, and Javadian (2014) develop a mathematical formulation for a single machine BP problem with release dates, aging effect and multi-maintenance activities to minimize makespan. The duration of the maintenance task is fixed in advance. To solve the problem, they propose two meta-heuristic approaches based on GA, Imperialist Competitive Algorithm, and a heuristic method.

Lu et al. (2018) take into account the unrelated parallel machines BP problem considering deteriorating jobs and maintenance activity to minimize makespan. The length of the maintenance task increases in accordance with its starting time. A mixed integer programming model is developed for the problem. Since the problem is NP-hard, they propose a hybrid of artificial bee colony (ABC) and Tabu Search to solve the problem.

Huang and Wang (2018) address a single machine BP problem with release dates and flexible preventive maintenance. A mathematical formulation and a two-stage solution method are developed for the problem.

Kong et al. (2020) propose a BRKGA-DE for the parallel machines BP problem taking into account the deterioration and learning effects as well as preventive maintenance. The processor/machine should be maintained after a specific number of batches have been completed.

Huang et al. (2020) formulate a mathematical model for a single machine BP problem with release dates, job families and flexible periodic preventive maintenance. An approach integrating rules with the GA is developed for the problem.

Our research work aims to minimize the total tardiness of a non-identical parallel machines BP problem with release dates and variable maintenance activity and it is denoted as $R_m|p-batch, MA, r_j|\sum_{j=1}^{n} t_j$ according to the standard machine scheduling classification. Based on the contributions of the aforementioned works, to the best of our knowledge, no research study previously has considered such a challenging machine scheduling problem. In this research, we first develop an MILP model to find optimal solutions for small-scale instances. Then, a constructive heuristic method is designed to schedule jobs on heterogeneous BP machines under the release dates and maintenance activity constraints. In addition, as the MILP formulation is not able to solve medium- and large-scale instances, two meta-heuristics based on SA and VNS are developed taking advantage of the proposed constructive heuristic to accelerate the seach. So as to validate the efficiency of the proposed solution methods, an experimental study is performed and the results of the proposed algorithms are compared with the results obtained by four meta-heuristics (PSO-GA by Beldar & Costa (2018), BKRGA-DE by Kong et al. (2020), ABC-TS by Lu et al. (2018), and IG by Arroyo et al. (2019)) adopted from the literature.

3. Problem Definition

BP problems have been widely studied in the scheduling literature due to their relevance to: the manufacturing of semiconductors (Uzsoy, 1994), heat treatments in metalworking (Lee, 1999), and cutting operations in the textile industry (Baker & Trietsch, 2009), to name a few. In BP, a processor/machine is able to process more than one job simultaneously. The completion time of the jobs in a batch is equal to the time when the last job of the batch is completed. Once the processing of a batch begins, the BP machine cannot stop; nor jobs can be added or removed from the batch.

Table 2
A numerical example, to illustrate the problem definition.

job	p_j	s_j	r_j	d_j
1	10	3	61	39
2	2	6	4	81
3	3	8	2	66
4	3	9	28	51
5	14	1	14	106
6	12	4	23	55
7	15	4	48	76
8	11	7	66	85
9	11	7	48	94
10	15	6	8	94
11	8	9	64	51
12	19	1	26	68

$B_1 = 10; B_2 = 11$
$LB = 28; UB = 88$
$bt_1 = 42; bt_2 = 53$

According to Beldar and Costa (2018), BP problems can be categorized according to two main parameters: the processing time required to finish the production of the batch, and the batch capacity.

The time required to process a batch can be determined as follows:

(I) p-batching problem: the processing time of the batch is equal to the longest processing time among the jobs allocated to the batch.
(II) s-batching problem: the processing time of the batch is equal to the sum of the processing times of the jobs allocated to the batch.
(III) The processing time of the batch is equal to a constant processing time.

In BP machine, the capacity of machines may be restricted by several factors:

(a) the number of jobs assigned to the batch is restricted by the maximum number of jobs that can be assigned to a machine.
(b) the number of jobs assigned to a batch depends on the weights of the jobs in the batch (i.e., volume, length, physical volume) and the capacity of the machines.
(c) the jobs assigned to a batch must respect both conditions (a) and (b).

A large number of research works in machine scheduling assume that processors are always available throughout the scheduling process. However, in the real-world production environment, the processors may become unavailable due to machine failures, maintenance tasks, tool replacement, etc. Unforeseen machine failures not only increase the production costs but also affect product quality. Hence, maintenance tasks play a significant role in decreasing the number of such failures. There are two main types of maintenance tasks: corrective maintenance (CM) and preventive maintenance (PM) (Avalos-Rosales et al., 2018). CM involves restoring the device to its desirable conditions, and is performed when a machine failure occurs. On the other hand, PM involves replacing, inspecting and cleaning machinery parts as required, thus preventing machine failure. Sometimes, keeping the device functioning until it completely fails can be extremely costly in terms of money, safety and time. Therefore, PM can dramatically decrease the probability of these unscheduled failures occurring, prolong the lifecycle of the device, and reduce the need for CM. Once the PM begins on a machine, the machine is not available for production purposes for a period of time, and no job can be performed by that machine, even if many production tasks need to be done. Consequently, production managers have to design their production schedule meticulously in order to keep down their costs while preventing the unanticipated

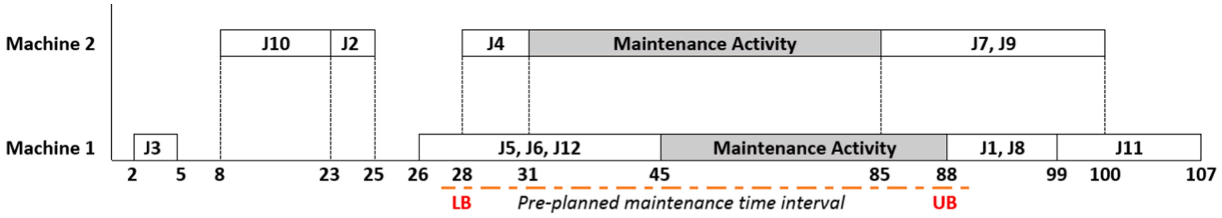


Fig. 2. Gantt chart, depicting a feasible solution.

unavailability of machines (Yoo & Lee, 2016). A rigorous combination of PM and job processing would help to create a better schedule.

In this research work, we assume that n jobs have to be processed on m unrelated BP machines. Let p_j denote the processing time of job $j = \{1, \dots, n\}$. All machines can process all jobs, and the jobs are not available at the beginning of the planning horizon (r_j). Each machine $l = \{1, \dots, m\}$ has a capacity B_l; each job $j = \{1, \dots, n\}$ has an associated weight s_j. The weights of the jobs in a batch cannot exceed the corresponding machine's capacity, and a machine can sequentially perform more than one batch. The processing time of a batch is equal to the processing time of the longest-to-process job in the batch. The batch processors must undergo pre-planned variable maintenance task to be accomplished within a certain time frame. As a matter of fact, because of the variable duration of the maintenance operation, a late maintenance beginning time would indicate a longer duration on one side, but a higher possibility of delay for the subsequent jobs on the other side. Contrarily, the earlier will be the maintenance beginning time the shorter will be its duration; consequently, the subsequent jobs will undergo a higher possibility of delay as well. MAs_l and MAc_l are decision variables and denote maintenance starting time and maintenance completion time on machine $l = \{1, \dots, m\}$ respectively. The duration of maintenance is a positive non-decreasing function of its starting time and calculated as $bt_l + \tan\alpha(MAs_l - LB)$, where α is a slope parameter and bt_l is maintenance base time on machine $l = \{1, \dots, m\}$. The maintenance task must be performed within a pre-planned time frame $[LB, UB]$, where LB and UB are the lower bound and upper bound of this range. Furthermore, $MAs_l \geqslant LB$ and $MAc_l \leqslant UB$. The completion time of job j is represented by C_j while $t_j = max\{0, C_j - d_j\}$ is the tardiness of job j and d_j is the corresponding due date. The objective is to incorporate the maintenance activity inside the specified time range in addition to finding a schedule capable of minimizing the total tardiness.

To illustrate, let us consider an example (see Table 2) of twelve jobs ($n = 12$) and two unrelated machines ($m = 2$). A feasible solution is depicted in the Gantt Chart shown in Fig. 2. The total tardiness for this test problem is equal to 160.

4. Mathematical formulation

In this section, an original MILP model is developed to address the proposed research problem. Indexes, parameters, decision variables, and the entire mathematical model are set out below:

Indexes:

$\{j \in J\}$ Sets of jobs.

$\{b \in B\}$ Sets of batches.

$\{l \in L\}$ Sets of machines.

Parameters:

n The number of jobs.

m The number of non-identical machines.

M A large number.

B_l Capacity of machine $l, l \in L, l=1, \dots, m$

N_l Number of batches on machine $l, l \in L, l=1, \dots, m, \sum_{l=1}^{m} N_l \leqslant n$.

p_j The processing time of job $j, j \in J, j=1, \dots, n$

s_j Size of job $j, j \in J, j=1, \dots, n$

r_j Release date of job $j, j \in J, j=1, \dots, n$

d_j Due date of job $j, j \in J, j=1, \dots, n$

LB Earliest starting time of the maintenance activity.

UB Deadline to accomplish the maintenance activity.

bt_l maintenance base time on machine $l, l \in L, l=1, \dots, m$

α Slope parameter of the flexible maintenance activity.

Decision variables:

$$X_{jb} = \begin{cases} 1 & \text{if job } j \text{ is assigned to batch } b \quad j \in J; b \in B, b = 1, \dots, n \\ 0 & \textit{Otherwise} \end{cases}$$

$$y_{bl} = \begin{cases} 1 & \text{if batch } b \text{ is processed on machine } l \quad b \in B; l \in L \\ 0 & \textit{Otherwise} \end{cases}$$

$$a_{bl} = \begin{cases} 1 & \text{if batch } b \text{ on machine } l \text{ is processed before the maintenance interval} \\ 0 & \textit{Otherwise} \end{cases}$$

P_{bl} Processing time of batch b on machine $l, b \in B, l \in L$

S_{bl} Starting time of batch b on machine $l, b \in B, l \in L$.

MAs_l Maintenance starting time on machine $l, l \in L$.

$MAsc_l$ Maintenance completion time on machine $l, l \in L$.

C_j Completion time of job $j, j \in J$.

t_j Tardiness of job $j \in J$.

Mathematical formulation:

$$\min \sum_{j=1}^{n} t_j \tag{1}$$

Subject to:

$$\sum_{b=1}^{n} X_{jb} = 1 j \in J \tag{2}$$

$$X_{jb} \leqslant y_{bl} l \in L \& j \in J \& b = N_{l-1} + 1, N_{l-1} + 2, \dots, N_l; N_0 = 0 \tag{3}$$

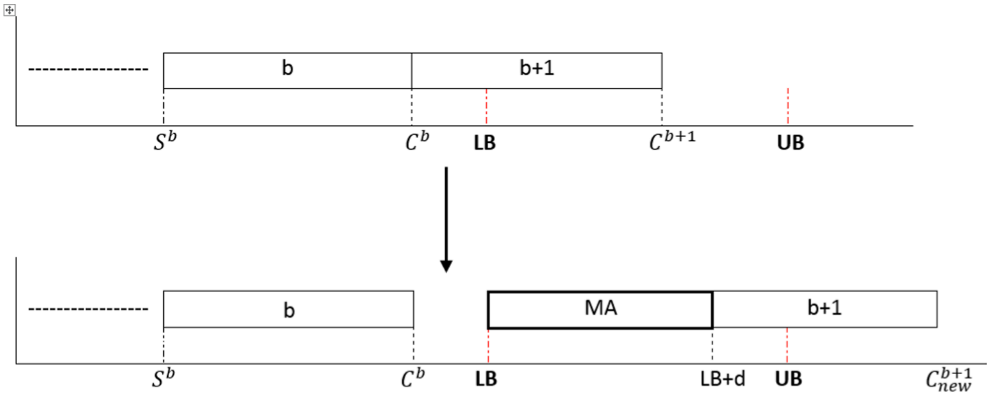


Fig. 3. The way of MA incorporation into the batches.

Table 3
A numerical example, to illustrate the proposed heuristic.

job	1	2	3	4	5	6	7
p_j	2	3	3	14	12	11	15
s_j	6	8	9	1	4	7	6
r_j	4	2	28	14	23	48	8
d_j	81	66	51	106	55	94	94
$B_1 = 10; B_2 = 11$ $LB = 28; UB = 88$ $bt_1 = 42; bt_2 = 53$							

$$\sum_{j=1}^{n} X_{jb} \geqslant y_{bl} l \in L \& b = N_{l-1}+1, N_{l-1}+2, \ldots, N_l; N_0 = 0 \tag{4}$$

$$y_{bl} \geqslant y_{b+1,l} l \in L \& b = N_{l-1}+1, N_{l-1}+2, \ldots, N_l - 1; N_0 = 0 \tag{5}$$

$$\sum_{b=N_{l-1}+1}^{N_l} a_{bl} \geqslant 1 l \in L; N_0 = 0 \tag{6}$$

$$a_{bl} \leqslant y_{bl} l \in L \& b = N_{l-1}+1, N_{l-1}+2, \ldots, N_l; N_0 = 0 \tag{7}$$

$$\sum_{j=1}^{n} X_{jb} \times s_j \leqslant B_l \times y_{bl} l \in L \& b = N_{l-1}+1, N_{l-1}+2, \ldots, N_l; N_0 = 0 \tag{8}$$

$$\begin{gathered} P_{bl} \geqslant p_j \times X_{jb} - M \times (1 - y_{bl}) \\ l \in L \& j \in J \& b = N_{l-1}+1, N_{l-1}+2, \ldots, N_l; N_0 = 0 \end{gathered} \tag{9}$$

$$\begin{gathered} S_{bl} \geqslant r_j \times X_{jb} - M \times (1 - y_{bl}) \\ l \in L \& j \in J \& b = N_{l-1}+1, N_{l-1}+2, \ldots, N_l; N_0 = 0 \end{gathered} \tag{10}$$

$$\begin{gathered} S_{bl} \geqslant S_{b-1,l} + P_{b-1,l} - M \times (1 - y_{b-1,l}) - M \times (1 - y_{bl}) \\ l \in L \& b(l = 1) = 2, 3, \ldots, N_1 \& b(l > 1) = N_{l-1}+1, N_{l-1}+2, \ldots, N_l \end{gathered} \tag{11}$$

$$\begin{gathered} S_{bl} + P_{bl} \leqslant MAs_l + M \times (1 - a_{b,l}) \\ l \in L \& b = N_{l-1}+1, N_{l-1}+2, \ldots, N_l; N_0 = 0 \end{gathered} \tag{12}$$

$$S_{bl} \geqslant MAc_l - M \times a_{b,l} l \in L \& b = N_{l-1}+1, N_{l-1}+2, \ldots, N_l; N_0 = 0 \tag{13}$$

$$MAs_l \geqslant LB l \in L \tag{14}$$

$$MAc_l \leqslant UB l \in L \tag{15}$$

$$MAc_l \geqslant MAs_l + bt_l + \tan\alpha \times (MAs_l - LB) l \in L \tag{16}$$

$$\begin{gathered} C_j \geqslant S_{bl} + P_{bl} - M \times (1 - X_{jb}) \\ l \in L \& j \in J \& b = N_{l-1}+1, N_{l-1}+2, \ldots, N_l; N_0 = 0 \end{gathered} \tag{17}$$

$$t_j \geqslant C_j - d_j j \in J \tag{18}$$

$$X_{jb} \in \{0, 1\} j \in J \& b \in B \tag{19}$$

$$y_{bl} \in \{0, 1\} b \in B \& l \in L \tag{20}$$

$$a_{bl} \in \{0, 1\} b \in B \& l \in L \tag{21}$$

$$P_{bl} \geqslant 0 b \in B \& l \in L \tag{22}$$

$$S_{bl} \geqslant 0 b \in B \& l \in L \tag{23}$$

$$MAs_l \geqslant 0 l \in L \tag{24}$$

$$MAc_l \geqslant 0 l \in L \tag{25}$$

$$C_j \geqslant 0 j \in J \tag{26}$$

$$t_j \geqslant 0 j \in J \tag{27}$$

The objective (1) is to minimize the total tardiness. Constraint (2) ensures that each job is assigned to only one batch. Constraints (3) and (4) state that if one batch is assigned to a machine, at least one job must be assigned to that batch and if one batch is not assigned to a machine, so no jobs are assigned to it. Constraint (5) guarantees that all active batches on a machine are ordered consecutively. Constraint (6) states that at least one batch on a machine is processed before the maintenance interval. Constraint (7) ensures that the batch on a machine which is supposed to be processed before the maintenance interval must have the same indexes as the chosen active batch on the machine. Constraint (8)

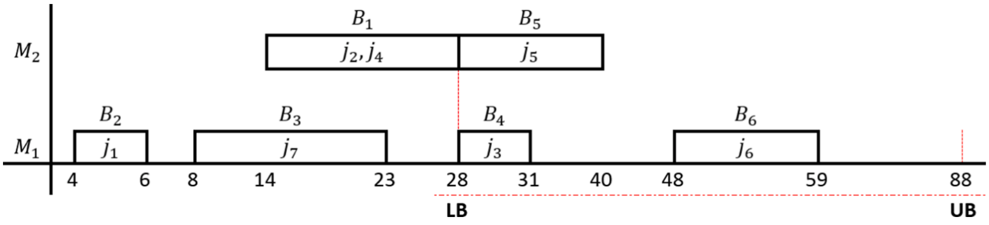


Fig. 4. Gantt chart of the solution obtained by the heuristic.

states that the total size of jobs assigned to a batch does not exceed the machine capacity. Constraint (9) defines the processing time of batch b on machine l which is the maximum processing time of jobs assigned to batch b. Constraint (10) guarantees that each job can be processed only after it is ready. Constraint (11) ensures that each batch can be started only after the previous one on the machine is finished. In case a batch on a machine is processed before the maintenance interval, constraint (12) states that its completion time must precede the maintenance starting time. In case a batch on a machine is processed after the maintenance interval, constraint (13) states that its starting time must follow the maintenance completion time. Constraints (14) and (15) fix bounds for maintenance starting and completion time respectively. Constraint (16) calculates maintenance duration. Constraint (17) states the completion time of a job. Constraints (18) define the tardiness of a job. Constraints (19)–(27)impose the binary and non-negativity nature of the decision variables.

5. Solution approaches

Since a single non-flexible maintenance activity to minimize the total tardiness has been proved to be NP-hard (Pinedo, 2012), the $R_m|p-batch, MA, r_j|\sum_{j=1}^{n} t_j$ problem, which is a major extension of the above problem, is strongly NP-hard as well. Our computational study revealed that mathematical formulations might not be able to find optimal solutions for very large instances; hence, a heuristic and two meta-heuristics are also proposed to solve this problem. We first discuss the constructive heuristic that could be of interest for real-time implementation, and then discuss meta-heuristic proposals.

5.1. The heuristic algorithm

Three major decisions are taken when scheduling BP machines with maintenance tasks: assigning jobs to batches, scheduling the formed batches on machines and assigning maintenance tasks on each machine. To do so, we develop a two-phase constructive heuristic approach. In the first phase, the approach tries to allocate jobs into the batches on different machines; in the second phase, the maintenance activities are positioned on the machines in a way that the total tardiness of jobs is minimized. The characteristics of the proposed heuristic are explained below:

First phase

Step 1. First, both the jobs and the machines are randomly ordered.

Step 2. The first job on the job list is assigned to the first batch and the formed batch is scheduled on the first machine on the machine list.

Step 3. The ensuing job is scheduled according to the following criteria:

Step 3.1. The job on the top of the list of remaining jobs is assigned separately to the existing batches having enough space on different machines.

Step 3.2. If the job could not be assigned in the previous step, a new batch is created separately on each machine and the job is assigned to that batch on the machine.

Step 4. Then both the makespan and the total tardiness are calculated for all possible combinations.

Step 5.Among all possible states, the one with the minimum total tardiness is selected. If there are some states with equal total tardiness, the one with minimum makespan is chosen. If their makespan is equal too, the machine with the smaller index is selected.

Step 6. If all jobs have been scheduled. Go to Step7. Otherwise, go to Step3.

Step 7. Stop algorithm.

Second phase

In this phase, in order to arrange the maintenance activity (MA) on each machine, we first start placing the maintenance activity between the first two batches from the right to the left on every single machine. Actually, the maintenance activity must be executed within the time

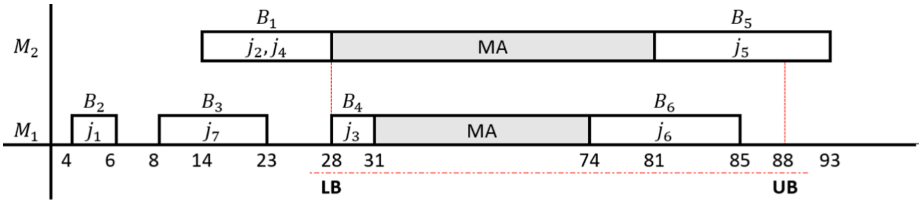


Fig. 5. Updated Gantt chart of the solution.

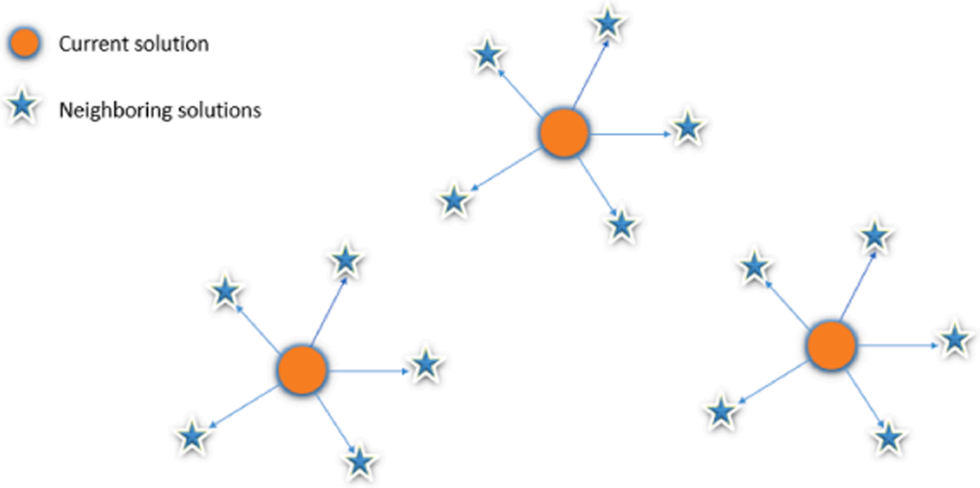


Fig. 6. A representation of a multi-start H + VNS.

interval $[LB, UB]$. If the maintenance activity is not within the time interval after incorporation into the two successive batches, then MA is scheduled between the two next batches and is checked in terms of the interval feasibility. This continues until the right position is found for the MA. Fig. 3 illustrates the incorporation of MA into the formed batches.

The starting time of the maintenance activity is calculated as $max\{C^b, LB\}$ where C^b is the completion time of the bth batch. On the other hand, the completion time of the maintenance activity is calculated as $max(\{C^b, LB\} + duration(d))$ which must be less than or equal to UB. Also, S^b in Fig. 3 is the starting time of batch b. An example of seven jobs and two machines is presented in Table 3 to better illustrate the proposed heuristic.

In the first phase, the jobs and machines are randomly ordered as $j_2, j_1, j_7, j_4, j_3, j_5, j_6$ and M_2, M_1. First job (j_2) on the jobs list is allocated into the first batch ($Batch_1 = \{j_2\}$) and then the batch is scheduled on the first machine on the machines list with $C_{max} = 5$. The next job on the jobs list is (j_1) which cannot be added to $Batch_1$ due to the capacity constraint ($s_2 + s_1 = 6 + 8 \geqslant B_2 = 11$). Hence, a new batch $Batch_2$ is made on each machine separately and its makespan and the total tardiness are calculated. In the state 1, $Batch_2 = \{j_1\}$ is scheduled on M_1 with $C_{max} = 6$ and $t_2 + t_1 = 0$ and also for the state 2, $Batch_2 = \{j_1\}$ is scheduled on M_2 with $C_{max} = 7$ and $t_2 + t_1 = 0$. Since the total tardiness of the state 1 and state 2 is equal, the state with the minimum makespan is chosen. Therefore, state 1 is selected. The next job on the jobs list (j_7) cannot be added to any existing batches due to the capacity constraint ($s_1 + s_7 = 6 + 6 \geqslant B_1 = 10$ and $s_2 + s_7 = 8 + 6 \geqslant B_2 = 11$). So, a new batch $Batch_3$ is created separately on each machine. In state 1, $Batch_3 = \{j_7\}$ is scheduled on M_1 with $C_{max} = 23$ and $t_2 + t_1 + t_7 = 0$ and also for state 2, $Batch_3 = \{j_7\}$ is scheduled on M_2 with $C_{max} = 23$ and $t_2 + t_1 + t_7 = 0$. Since the tardiness and the makespan of state 1 and state 2 are equal, the state with the smaller machine index is chosen; here, it is state 1. The next job on the jobs list (j_4) can be added to all the existing batches. Therefore, in state 1, j_4 is incorporated into $Batch_2 = \{j_1, j_4\}$ on M_1 and the makespan and the tardiness are modified as $C_{max} = 43$ and $t_2 + t_1 + t_4 + t_7 = 0$. In state 2, j_4 is assigned to $Batch_3 = \{j_7, j_4\}$ on M_1 with $C_{max} = 29$ and $t_2 + t_1 + t_7 + t_4 = 0$. In state 3,$j_4$ is assigned to $Batch_1 = \{j_2, j_4\}$ on M_2 with $C_{max} = 28$ and $t_2 + t_4 + t_1 + t_7 = 0$. On the other hand, a new batch $Batch_4$ is made on each machine separately. In state 4, $Batch_4 = \{j_4\}$ is scheduled on M_1 with $C_{max} = 37$ and $t_2 + t_1 + t_7 + t_4 = 0$ and also for state 5, $Batch_4 = \{j_4\}$ is scheduled on M_2 with $C_{max} = 28$ and $t_2 + t_4 + t_1 + t_7 = 0$. Of all the five states, 3 and 5 on M_2 have the minimum tardiness and makespan. Since state 3 has fewer batches (three formed batches) than state 5 (four formed batches), state 3 is chosen. If the next job on the jobs list (j_3) is added to any existing batches one by one, the total size of jobs in each of the batches exceeds the batch capacity. Thus, a new batch $Batch_4$ is individually formed on each machine. In state 1, $Batch_4 = \{j_3\}$ is scheduled on M_1 with $C_{max} = 31$ and $t_2 + t_4 + t_1 + t_7 + t_3 = 0$ and also for state 2, $Batch_4 = \{j_3\}$ is scheduled on M_2 with $C_{max} = 31$ and $t_2 + t_4 + t_3 + t_1 + t_7 = 0$. As the makespan and the total tardiness of both states are identical, the state with the smaller machine index, state 1, is chosen. The next job on the jobs list (j_5) can be incorporated into both $Batch_2$ and $Batch_3$. Hence, in both states, batch modification is carried out as $Batch_2 = \{j_1, j_5\}$ and $Batch_3 = \{j_7, j_5\}$ separately. In the first state, the makespan is equal to 53 and the total tardiness ($t_2 + t_4 + t_1 + t_5 + t_7 + t_3$) is equal to 2, while in the second state, the makespan and the total tardiness is 41 and 0 respectively. Two other states, which is according to the creation of a new batch ($Batch_5$) on each machine, should be taken into account. In the third state, $Batch_5 = \{j_5\}$ is scheduled on M_1 with $C_{max} = 43$ and $t_2 + t_4 + t_5 + t_1 + t_7 + t_3 = 0$ and also for the fourth state, $Batch_5 = \{j_5\}$ is scheduled on M_2 with $C_{max} = 40$ and $t_2 + t_4 + t_1 + t_7 + t_3 + t_5 = 0$. Of all the four states, state 4 has both the minimum makespan and the minimum total tardiness. The last job on the jobs list (j_6) can be added to $Batch_5 = \{j_5, j_6\}$ on M_2 as state 1. The makespan and the total tardiness are 60 and 5 respectively ($t_2 + t_4 + t_5 + t_6 + t_1 + t_7 + t_3 = 5$). On the other hand, a new batch ($Batch_6$) is made on each machine individually. In state 2, $Batch_6 = \{j_6\}$ is scheduled on M_1 with $C_{max} = 59$ and $t_2 + t_4 + t_5 + t_1 + t_7 + t_3 + t_6 = 0$ and also for state 3, $Batch_6 = \{j_6\}$ is scheduled on M_2 with $C_{max} = 59$ and $t_2 + t_4 + t_5 + t_6 + t_1 + t_7 + t_3 = 0$. Since the makespan and the total tardiness for states 2 and 3 is the same, the one with the smaller machine index (state 2) is selected. Fig. 4 shows the Gantt chart of the solution obtained by heuristic.

After all jobs have been scheduled, the second phase begins. From the right to the left, MA is positioned between two consecutive batches on each machine. Hence, on M_1, we first place the MA between $Batch_4$ and $Batch_6$ and calculate MAs_1 and MAc_1 to determine whether both

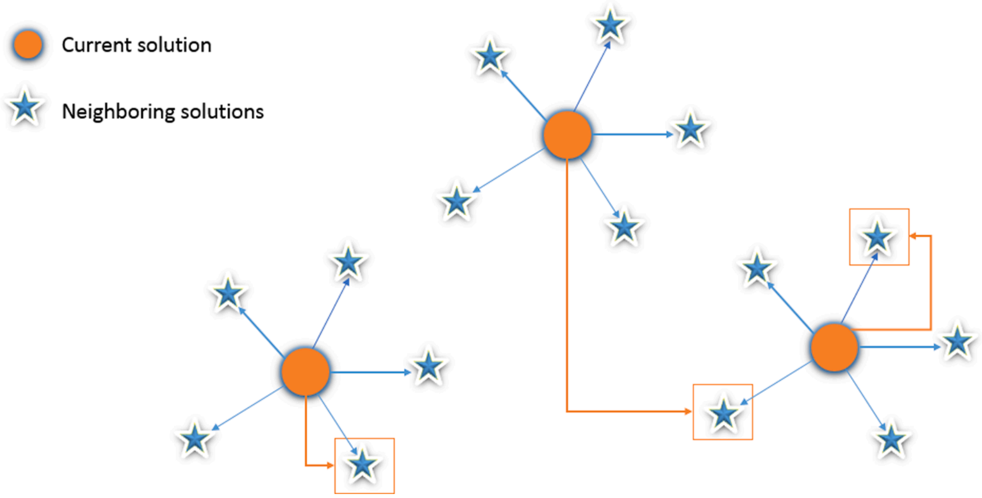


Fig. 7. A representation of the new solutions.

Table 4
Data generation.

Parameters	*Small*	*Medium*	*Large*
n	$U[12, 20]$	$U[21, 50]$	$U[51, 100]$
m	2	2	2
		4	4
			6
p_j	$U[1, 20]$	$U[1, 20]$	$U[1, 20]$
	$U[1, 50]$	$U[1, 50]$	$U[1, 50]$
r_j	$0.5 \times U[0, K]$	$0.5 \times U[0, K]$	$0.5 \times U[0, K]$
	$0.75 \times U[0, K]$	$0.75 \times U[0, K]$	$0.75 \times U[0, K]$
s_j	$U[1, 10]$	$U[1, 10]$	$U[1, 10]$
	$U[4, 10]$	$U[4, 10]$	$U[4, 10]$
d_j	$(rand(0, 1) \times 0.5 + 0.25) \times K$	$(rand(0, 1) \times 0.5 + 0.25) \times K$	$(rand(0, 1) \times 0.5 + 0.25) \times K$
B_l	10, 11	10, 11; $m = 2$	10, 11; $m = 2$
		10, 12, 13, 11; $m = 4$	10, 12, 13, 11; $m = 4$
			10, 12, 14, 11, 15, 13; $m = 6$
LB	$0.2 \times K$	$0.2 \times K$	$0.2 \times K$
UB	$LB + 3 \times P_{up}$	$LB + 3 \times P_{up}$	$LB + 3 \times P_{up}$
bt_l	$U[40, 60]$	$U[40, 60]$	$U[40, 60]$
	$U[100, 150]$	$U[100, 150]$	$U[100, 150]$
Overall states	8	16	24

values are within the time interval $[LB = 28, UB = 88]$. $MAs_1 = max\{C_4 = 31, LB = 28\} = 31$ and $MAc_1 = MAs_1 + bt_1 + \tan\alpha \times (MAs_1 - LB) = 31 + 42 + 0.002 * (31 - 28) = 74$. As $MAs_1 = 31 \geqslant LB = 28$ and $MAc_1 = 74 \leqslant UB = 88$, thus this is the appropriate position for MA on M_1. Similar steps are taken to position the MA on M_2. The MA is incorporated between $Batch_1$ and $Batch_5$ and MAs_2 and MAc_2 are calculated. $MAs_2 = max\{C_1 = 28, LB = 28\} = 28$ and $MAc_2 = MAs_2 + bt_2 + \tan\alpha \times (MAs_2 - LB) = 28 + 53 + 0.002 * (28 - 28) = 81$. Since both values are within the time interval, this is the right position for MA on M_2. The updated Gantt chart of the solution after MA assignment to both machines is shown in Fig. 5.

5.2. Meta-heuristic algorithms

Meta-heuristics are generally categorized as either single-start algorithms or multiple-start algorithms. In the former case, the algorithm begins with an initial solution and iteratively moves from the current solution to another feasible solution to find a better solution. The VNS, TS, IG, and SA belong to this classification. In the latter, the algorithm starts with multiple solutions and iteratively makes changes to the solutions to improve their quality. PSO, GA, and ABC belong to this classification. Since, VNS and SA have shown great performance in solving scheduling issues (Kumar Manjeshwar, Damodaran, & Srihari, 2009; Wang & Chou, 2010; Lei & Guo, 2011; Damodaran & Vélez-Gallego, 2012; Bilyk, Mönch, & Almeder, 2014; Tan, Mönch, & Fowler, 2018; Pacheco, Porras, Casado, & Baruque, 2018; Ying & Lin, 2020; Wu et al., 2021; Lin, Cheng, Pourhejazy, & Ying, 2021), we propose H + VNS and H + SA search techniques that take advantage of the heuristic search H proposed in Section 5.1 to improve the search technique. The heuristic H generates an initial population that will be used by the meta-heuristics to obtain optimized solutions in a more efficient way.

5.2.1. *Variable Neighborhood Search*

VNS is a single-start meta-heuristic which was first proposed by Mladenovic and Hansen (1997), and then, more extended versions were posited by Hansen and Mladenovic (2001) and Hansen, Mladenovic, and Pérez (2008). It makes systematic changes to neighborhoods in two steps: descent step and shaking (perturbation) step (Hansen et al., 2008). In the descent step, VNS focuses the search spotlight on a local area by intensifying the knowledge that an incumbent good candidate solution is discovered in this particular area; whereas, in the shaking step, VNS globally probes the search area to find other unseen solutions. VNS is an approach which attempts to create a fine balance between the intensification step (descent step) and diversification step (shaking step). The basic VNS begins with an initial solution which is either randomly produced or constructively produced by heuristics. Then, during the algorithm process, a solution is randomly chosen from a predefined neighborhood search structure N_k ($k = 1, \dots, k_{max}$) in the perturbation step and considered as an initial candidate for the local search procedure. The local search procedure probes the search space by applying the neighborhood structure. Each solution is compared with the solution obtained from the perturbation step and the best solution is recorded. The recorded solution is compared with the global best solution. If an enhancement occurs, the best solution ever found is updated and the local search continues with $k = 1$. Otherwise, in order to escape from the local optimum, the algorithm carries on the search with another neighborhood structure ($k = k + 1$) to explore other regions to find a better solution. The algorithm process is repeated until the termination condition is satisfied.

5.2.1.1. Neighborhood structures. In this research, six neighborhood structures are applied in the local search procedure in order to find a better solution. The example shown in Table 3, and the solution obtained from the heuristic in the first phase before the maintenance assignment as shown in Fig. 4 are intended to better illustrate how each neighborhood structure functions. It is worth mentioning that after each possible move, the second phase of the heuristic is performed.

Swap jobs01. The job in ath position of bth batch on cth machine is swapped with the job in kth position of gth batch on hth machine. If bth and gth batches have enough space, the move is feasible. Otherwise the move is unfeasible and it is rejected from the neighborhood. For instance, according to the solution obtained from the heuristic, a possible move is a swap between j_3 in B_4 on *Machine*1 and j_5 in B_5 on *Machine*2.

Swap jobs02. Two adjacent batches on the same machine are chosen randomly and one job is chosen randomly from each of them and their positions are swapped. For instance, B_1 and B_5 are selected and a swap between j_4 and j_5 is made.

Insertion01. The job in ath position of bth batch on cth machine is removed and inserted into kth position (empty position) of gth batch on hth machine, if the gth batch has enough space, the move is feasible. Otherwise the move is unfeasible and it is rejected from the neighborhood. For example, j_5 in B_5 is removed from *Machine*2 and inserted into B_2 on *Machine*1.

Insertion02. The job in ath position of bth batch on cth machine is removed and inserted into a new empty batch which is created on hth machine. For instance, j_5 in B_5 is removed from *Machine*2 and inserted into a new batch on *Machine*1.

Swap batches01. The bth batch on cth machine is swapped with the gth batch on hth machine, if this change does not violate the cth and hth machines' capacity. As an example, B_3 on *Machine*1 is swapped with B_5 on *Machine*2.

Swap batches02. Two adjacent batches on the same machine are chosen randomly and their positions are swapped. For example, B_1 and B_5 on *Machine*2 are swapped.

5.2.1.2. Shaking procedure. The proposed H + VNS employs four different shaking strategies to perturb the incumbent solution in order to escape from the local optimum. One of the strategies is randomly chosen at the shaking stage. If the selected strategy does not provide a feasible move, then another strategy is randomly selected.

Shake1. A batch from the machine with the maximum makespan is removed and inserted into the machine with the minimum makespan.

Shake2. A batch from the ath machine and a batch from the kth machine are randomly chosen and merged together. This move is acceptable if the capacity of the new merged batch does not exceed the machine capacity.

Shake3. Two different bathes are randomly chosen on the same machine. A cut point (cp) is selected on the first batch ($2 \leqslant cp \leqslant n_j - 1$), where n_j is the number of jobs in the first batch. The jobs after cp from the first batch are removed and added into the second batch.

Shake4. If no improvement occurs after a certain number of iterations, all the batches are destroyed and jobs are relocated to the machines based on the heuristic described in Section 5.1.

5.2.1.3. Multi-start VNS. In order to enjoy the benefit of both intensification capacity of the single-start meta-heuristics and the diversification capacity of the multiple-start meta-heuristics, we develop a multiple-start H + VNS for the problem under study. As VNS is a single-start meta-heuristic, a mechanism is needed to transform it into a multiple-start H + VNS. To do so, a population of initial solutions is obtained by the heuristic developed in Section 5.1 and its best solution is recorded as the global solution. At each iteration, each of the solutions produces a number of neighboring solutions in the local search procedure based on the neighborhood structures discussed in Section 5.2.1.1. Fig. 6 shows an example of a population with three members and their five neighboring solutions. All the solutions found in the local search for each member of the population are stored on a list of new solutions in the size of (number of population * number of moves). As depicted in Fig. 6, the list comprises 15 new solutions (3*5). The new solutions are sorted from the best to the worst, and as many of the best solutions as the population size (=3) are chosen (see Fig. 7). The new solutions are compared with the current solutions and the current solutions are updated. The best of these updated solutions is selected and compared with the global best solution. If the updated solution is better than the global best solution, the global best solution is updated and $l = 1$. Otherwise, $l = l + 1$. This process continues until the stopping criterion is met. The pseudo-code of the proposed H + VNS is presented in Algorithm 1;

Algorithm 1. Pseudo-code of H + VNS

```
Input: Population size, Number of Moves, A set of neighborhood
       structures $N_l, l = 1, 2, .., l_{max}$
Result: Global best solution ($G_{best}$)
Generate Initial Solutions;
$G_{best} = +Inf$;
do
    $l \leftarrow 1$;
    do
        for $i \leftarrow 1$ to $PopulationSize$ do
            $s_i \leftarrow shake(InitialSolution_i, N_l)$ ;
            for $j \leftarrow 1$ to $NumberofMoves$ do
                $list_i^j \leftarrow localsearch(s_i)$ ;
            end
        end
        Sort $list_i^j$ based on the solutions ranging from best to worst;
        $s_k^{"} \leftarrow$ Choose from the sorted list in the size of population size
         Where, k=1,.., $PopulationSize$;
        for $k \leftarrow 1$ to $PopulationSize$ do
            if $s_k^{"} < InitialSolution_k$ then
                $InitialSolution_k \leftarrow s_k^{"}$;
            end
        end
        $LocalBest \leftarrow \min_{i=1,..,PopulationSize} InitialSolution(i)$;
        if $LocalBest < G_{best}$ then
            $G_{best} \leftarrow LocalBest$;
            $l \leftarrow 1$;
        else
            $l \leftarrow l + 1$;
        end
    while $l \leq l_{max}$;
while Stopping criterion is not met;
```

5.2.1.4. Stopping criterion. The aforementioned process is iterated until the termination condition according to the maximum time specified for each test problem is met.

5.2.2. Heuristic + Simulated Annealing (H + SA)

Similarly to VNS, SA is a single-start meta-heuristic. It was first introduced by Kirkpatrick, Gelatt, and Vecchi (1983) and Cerny (1985). It was inspired by the analogy between the physical annealing of metals and the process of the searching for the optimal solution to a combinatorial optimization problem (Damodaran & Vélez-Gallego, 2012). SA begins with an initial solution, then during the search process of the algorithm, it generates a neighboring solution at each iteration according to the mechanism considered for the neighborhood generation. If the objective function value of the neighboring solution is better than the current solution, the current solution is updated. Otherwise, in order to avoid being trapped in a local optimum, the algorithm accepts the bad (non-improving) neighboring solution with a certain probability which decreases as the algorithm proceeds. The algorithm process is iterated until the stopping criterion is met.

5.2.2.1. The proposed H + SA. As depicted in Algorithm 2, the proposed H + SA has two main loops. In the outer loop, a population of initial solutions is produced by the heuristic H discussed in Section 5.1. The best solution is recorded as B_{IS} and the difference between the best objective function value and the worst one among the population is recorded as *DF*. In addition, the maximum number of iterations and the maximum threshold for restarting the process are defined as Max_{Iter} and R_{Mtd}. Max_{Iter} is set to the maximum primary iteration (P_{MI}) which is given as an input of the algorithm and R_{Mtd} is set to $\beta \times Max_{Iter}$. Moreover, the best solution before restarting the process is defined as G_{best} and set to a large value. In the inner loop, which continues till $Max_{Iter} < R_{Mtd}$,the following steps are taken:

Step 1. Temperature (T) and $\alpha$are set as $\frac{-DF}{\log 0.95}$ and $(\frac{0.1}{T})^{\frac{1}{Max_{Iter}}}$ respectively.

Step 2. For a number of iterations equal to Max_{Iter}, SA is performed. At each iteration, one of the neighborhood structures explained in Section 5.2.1.1 is randomly used, and a neighboring solution is found and stored as the temporary solution (S_{temp}). If the temporary solution (S_{temp}) is better than the current solution (B_{IS}), then the current solution is updated.The current solution is also recorded as the secondary solution (S_{sec}). Otherwise, a random number between 0 and 1 is generated. If the number is less than or equal to $\exp((-1 \times (S_{temp} - B_{IS})/T))$, the current solution is updated. At each iteration, T is updated as $\alpha \times T$. After SA stops, the secondary solution (S_{sec}) and B_{IS} are obtained as outputs.

Step 3. If S_{sec} is better than B_{IS}, the B_{IS} is updated. This condition is checked to ensure the best output during the current series of iterations.

Step 4. If B_{IS} is better than In_{best} (the best solution found in the inner loop), DF and Max_{Iter} are updated as $DF = t \times DF$ and $Max_{Iter} = r \times Max_{Iter}$, where t and r take values greater than 1. Otherwise, DF and Max_{Iter} are updated as $DF = t \times DF$ and $Max_{Iter} = r \times Max_{Iter}$, where t and r take values less than 1.

Step 5. If In_{best} is better than the global best solution (G_{best}), G_{best} is updated.

Algorithm 2. Pseudo-code of H + SA

```
Input: Maximum primary iteration ($P_{MI}$), Population Size, n, m
Result: Global best solution ($G_{best}$)
initialization;
$G_{best} = +Inf$;
do
    Generate initial population by running heuristic H
while Stopping criterion is not met;
;
Record the best solution as $B_{IS}$;
$DF$ = the best solution - the worse solution;
$Max_{Iter} = P_{MI}$;
$R_{Mtd} = \beta \times Max_{Iter}$;
$In_{best} = +Inf$;
do
    $T = \frac{-DF}{\log 0.95}$;
    $\alpha = (\frac{0.1}{T})^{\frac{1}{Max_{Iter}}}$;
    for $i \leftarrow 1$ to $Max_{Iter}$ do
        Generate a neighbor solution and store it as ($S_{temp}$);
        if ($S_{temp} < B_{IS}$) then
            $B_{IS} \leftarrow S_{temp}$;
            $S_{sec} \leftarrow S_{temp}$;
        else
            if $rand(0,1) \leq \exp\left((-1 \times (S_{temp} - B_{IS})/T)\right)$ then
                $B_{IS} \leftarrow S_{temp}$;
            end
        end
        $T = \alpha \times T$;
    end
    if $S_{sec} < B_{IS}$ then
        $B_{IS} \leftarrow S_{sec}$
    end
    ;
while $Max_{Iter} < R_{Mtd}$;
if $B_{IS} \leq In_{best}$ then
    $In_{best} \leftarrow B_{IS}$;
    $DF = t \times DF$; $t > 1$;
    $Max_{Iter} = r \times Max_{Iter}$; $r > 1$;
else
    $DF = t \times DF$; $t < 1$;
    $Max_{Iter} = r \times Max_{Iter}$; $r < 1$;
end
if $In_{best} < G_{best}$ then
    $G_{best} \leftarrow In_{best}$;
end
```

6. Data generation and computational results

In this section, a set of test instances were randomly produced in order to compare the performance of the proposed solution approaches. The detailed description of the way of data generation, the computational time and the obtained numerical results are presented in the following sub-sections.

6.1. Test instances

In order to assess the effectiveness of the proposed solution methods, a vast range of test instances were randomly produced according to the following parameters: the number of jobs (n), the number of non-identical machines (m), processing time of jobs (p_j), the release date of jobs (r_j), the size of jobs (s_j), the due date of jobs (d_j), the maximum capacity of the machine (B_l), the earliest starting time of the maintenance activity (LB), the deadline for accomplishing the maintenance activity (UB), and the maintenance base time on each machine (bt_l). Three different categories of test instances (small, medium, large) were used as shown in Table 4. K is set as $1.15 \times \sum_{j=1}^{n} p_j$ and P_{up} is equal to the upper bound of the interval considered for processing time of jobs, for instance, in the range of $[1, 20]$, P_{up} is set to be 20. The slope parameter α is set to be 0.15. Each state is randomly repeated five times to analyse different types of scenarios. Hence, the total number of instances to be solved by means of each solution method is equal to $(8 + 16 + 24) \times 5 = 240$.

6.2. Execution time

The termination condition of the proposed solution approaches is according to the execution time (ET). ET increases as the number of jobs increases, thus ET is defined as a function of number of jobs (n):

$$ET = \frac{n \times Max_{Time}}{n_{upper}} \tag{28}$$

Where Max_{Time} and n_{upper} are the maximum allowable execution time and the upper value of the interval associated with the number of jobs, respectively; for instance, n_{upper} is equal to 50 in the interval considered for the medium-sized test problems $[21, 50]$. Max_{Time} has been determined to be 30, 90, 180 s for small, medium, and large-scale test instances respectively. All the proposed algorithms were coded in Visual Basic programming language. All the test instances were run on a Core i5 laptop with 1.7 GHz CPU and 4 GB RAM.

6.3. Experimental results

A Relative Percentage Deviation (RPD) performance indicator was applied for the comparison of the six evaluated methods. CPLEX was capable of optimally solving all the small-scale test problems and some medium-scale test problems (37 test instances out of 80). As CPLEX did not converge to an optimal solution even after performing for several hours on remaining medium-scale test instances, CPLEX was terminated after running for 1800 s and the best-known solution was recorded. The RPD values were calculated based on the global optimum achieved using ILOG CPLEX 20.1 as shown in Eq. (29). However, the RPD values for medium- and large-scale test problems were calculated as shown in Eq. (30) according to the best solution obtained by the proposed solution approaches for each test problem. Each solution method was run five times, then the best and the average values were reported.

$$RPD = \frac{algorithm_{solution} - global_{optimum}}{global_{optimum}} \times 100 \tag{29}$$

$$RPD = \frac{algorithm_{solution} - best_{solution}}{best_{solution}} \times 100 \tag{30}$$

The computational results are shown in Tables 5–10 for all the test instances. The results for the small-sized instances are shown in Table 5. Column 1 and Column 2 present the test instances and the size of each test instance, respectively. Columns 3–14 show the best and the average results of each algorithm over five runs. The execution time for each test instance is listed in Column 15. The best or optimal value and the computational time obtained by CPLEX are reported in Columns 16–17. The RPD values for the best (RPD_{Best}) and the average ($RPD_{Average}$) obtained by each solution approach are depicted in Columns 18–29. The similar format is applied for Tables 6–10. The average RPD values for the best depicted in Table 5 demonstrate the efficiency of both H + SA and H + VNS over PSO-GA, BKRGA-DE, IG, and ABC-TS for the small-sized test instances. The average RPD values of the best (RPD_{Best}) less than 1% confirm that both H + SA and H + VNS have been appropriately designed for the research problem under investigation. Moreover, the results presented in Tables 6–10 indicate the performance of the solution methods applied to medium- and large-sized test problems. It can be observed that, for all the test instances, H + SA outperforms other solution approaches. Moreover, Fig. 8 also shows the superiority of H + SA over other solution methods based on both the average RPD_{Best} and $RPD_{Average}$ achieved by each solution technique for the three different categories of test instances. In addition, it can be inferred from the results of RPDs that the problems with a combination of $P_2 \rightarrow U[1, 50]$ and $S_2 \rightarrow U[4, 10]$ and also a combination of $P_2 \rightarrow U[1, 50]$ and $r_2 \rightarrow 0.75 \times U[0, K]$ have difficulty in finding a high-quality solution.

The MINITAB 19 commercial package was employed to obtain statistical results from the entire set of outputs of the algorithms. As the normality test was not fulfilled over the RPD results, a Kruskal–Wallis non-parametric test on medians (Beldar, Framinan, & Ardakani, 2019) was considered to be the most suitable statistical method to compare the solution approaches for different categories of test instances. Tables 11–13 display the results of the non-parametric test for each category of test instances. The results demonstrate that there was statistically significant difference among the performance of different solution methods (The adjusted P-Value is less than 0.05) for each category. Furthermore, the Box-plot diagram at 95% confidence level shown in Fig. 9 highlights that the difference among the different solution approaches is significant. According to the Z rank in Tables 11–13 and Box-plot diagram for different solution methods, H + SA and H + VNS are the most promising solution approaches. Therefore, similarly being carried out by Beldar et al. (2019), a post hoc Mann–Whitney non-parametric pairwise test (Mann & Whitney, 1947) was performed in order to make a comparison between H + SA and H + VNS to discover the solution approach with the best performance. Table 14 shows that there is a significant statistical difference (The adjusted P-Value is equal to 0.05) between H + SA and H + VNS. Hence, H + SA outperforms H + VNS.

The convergence status of six meta-heuristics are discussed so as to further investigate their efficiency. The outputs of three different test instances are chosen to form the convergence curve of the proposed solution methods. Fig. 10 depicts the convergence curves of sixsolution approaches for a particular test instance with $n = 27$ and $m = 2$. As it can be drawn from Fig. 10, H + SA enjoys superiority over other solution techniques in terms of the convergence speed, but the performance of H + SA and H + VNS is relatively close to one another with respect to the quality of solution. Fig. 11 shows the convergence curves of each method for the test instance with $n = 40$ and $m = 4$. It displays that H + SA is clearly superior to other solution approaches in terms of both the solution quality and the convergence speed. Similarly, Fig. 12 shows the convergence status of different solution methods for a particular test instance with $n = 72$ and $m = 2$. As it is clear, the convergence speed of H + SA is better than other methods, but its quality of solution is close to that of the H + VNS. Generally speaking, it can be inferred that H + SA has superiority over other solution methods with respect to both the solution quality and convergence speed.

Table 5
Small-scale test problems: Best and Average tardiness, ET in seconds, Optimal and Elapse Time of CPLEX, and the Best and the Average RPD values

		H + SA		H + VNS		PSO-GA		BKRGA-DE		IG		ABC-TS
Test problem	(n,m)	Best	Average	Best	Average	Best	Average	Best	Average	Best	Average	Best
n1m1P1S1r1-01	(19,2)	158	158.0	158	158.0	158	158.0	158	160.8	159	160.6	158
n1m1P1S1r1-02	(15,2)	198	199.4	198	199.8	201	203.4	201	206.4	200	201.8	199
n1m1P1S1r1-03	(19,2)	195	196.6	197	198.6	200	202.4	197	208.2	207	208.2	204
n1m1P1S1r1-04	(16,2)	214	214.8	215	216.8	220	221.2	219	228.0	215	219.2	220
n1m1P1S1r1-05	(12,2)	160	160.0	160	160.0	160	160.0	160	160.0	160	160.0	160
n1m1P1S1r2-01	(16,2)	548	548.0	548	548.0	548	551.2	548	552.6	548	550.0	548
n1m1P1S1r2-02	(15,2)	424	424.0	424	424.0	424	424.0	424	428.8	424	424.2	424
n1m1P1S1r2-03	(15,2)	437	437.0	437	437.0	438	438.6	437	437.4	437	437.6	437
n1m1P1S1r2-04	(19,2)	184	184.0	184	184.0	184	185.2	184	187.2	186	187.6	191
n1m1P1S1r2-05	(18,2)	101	101.0	101	101.0	103	104.8	101	102.4	103	107.6	106
n1m1P1S2r1-01	(18,2)	93	93.0	93	93.6	103	103.0	93	97.6	104	106.0	96
n1m1P1S2r1-02	(19,2)	166	166.0	166	166.0	169	169.2	168	177.2	172	174.0	174
n1m1P1S2r1-03	(14,2)	228	230.4	228	229.2	237	238.4	228	235.8	231	232.2	231
n1m1P1S2r1-04	(13,2)	97	97.0	97	97.0	97	99.6	97	97.0	97	98.2	97
n1m1P1S2r1-05	(17,2)	113	113.2	113	116.4	120	120.8	113	122.6	120	122.8	125
n1m1P1S2r2-01	(17,2)	442	442.0	442	442.0	442	442.0	442	444.2	442	442.4	442
n1m1P1S2r2-02	(13,2)	273	273.6	273	273.0	275	275.0	274	274.4	273	273.0	274
n1m1P1S2r2-03	(17,2)	311	311.0	311	311.0	311	313.8	311	313.0	311	315.4	311
n1m1P1S2r2-04	(17,2)	442	442.0	442	442.0	442	442.0	442	443.6	442	442.8	442
n1m1P1S2r2-05	(17,2)	412	412.0	412	412.0	412	412.2	412	412.8	412	414.4	414
n1m1P2S1r1-01	(19,2)	483	484.8	483	486.2	495	499.4	508	523.6	507	512.8	518
n1m1P2S1r1-02	(16,2)	697	697.0	697	697.0	697	713.2	711	718.2	711	712.4	718
n1m1P2S1r1-03	(19,2)	232	232.0	232	232.6	235	235.4	232	235.0	232	236.2	235
n1m1P2S1r1-04	(14,2)	842	842.0	842	842.0	842	842.0	842	842.0	842	842.0	842
n1m1P2S1r1-05	(13,2)	595	595.0	595	595.0	595	597.0	595	596.0	595	595.0	595
n1m1P2S1r2-01	(15,2)	1027	1027.0	1027	1027.2	1029	1032.0	1029	1083.6	1027	1027.4	1045
n1m1P2S1r2-02	(18,2)	1279	1279.0	1279	1279.0	1279	1279.0	1279	1280.0	1279	1280.0	1289
n1m1P2S1r2-03	(19,2)	388	388.0	388	388.0	388	388.2	388	388.2	388	389.4	388
n1m1P2S1r2-04	(18,2)	1408	1408.0	1408	1408.0	1408	1410.4	1408	1408.0	1408	1412.4	1411
n1m1P2S1r2-05	(15,2)	712	713.4	712	714.8	719	723.0	724	729.0	712	719.0	729
n1m1P2S2r1-01	(14,2)	795	795.0	795	795.0	795	801.4	795	795.8	795	795.4	802
n1m1P2S2r1-02	(19,2)	600	600.0	600	600.0	600	602.6	600	602.0	600	604.6	606
n1m1P2S2r1-03	(13,2)	593	593.0	593	594.0	593	597.6	593	598.4	593	593.0	608
n1m1P2S2r1-04	(13,2)	334	334.0	334	334.0	334	334.0	334	334.0	334	334.0	339
n1m1P2S2r1-05	(18,2)	189	189.0	189	189.0	189	190.6	189	189.0	189	189.0	189
n1m1P2S2r2-01	(12,2)	925	925.0	925	925.0	925	926.6	925	930.0	925	925.0	925
n1m1P2S2r2-02	(15,2)	885	885.0	885	885.0	886	886.0	886	888.0	885	885.4	885
n1m1P2S2r2-03	(14,2)	920	920.0	920	920.0	920	920.2	920	920.6	920	920.0	921
n1m1P2S2r2-04	(19,2)	948	948.0	948	948.0	948	951.6	948	955.8	954	958.8	969
n1m1P2S2r2-05	(16,2)	402	402.0	402	403.0	402	403.0	407	407.2	404	405.2	407
Average												

ABC-TS			CPLEX	RPD_{Best}							$RPD_{Average}$				
Average	ET(sec)	Optimal	Elapsed Time(sec)	H + SA	H + VNS	PSO-GA	BKRGA-DE	IG	ABC-TS	H + SA	H + VNS	PSO-GA	BKRGA-DE	IG	ABC-TS
160.8	28.5	158	327.52	0.00	0.00	0.00	0.00	0.63	0.00	0.00	0.00	0.00	1.77	1.65	1.77
206.2	22.5	192	261.63	3.13	3.13	4.69	4.69	4.17	3.65	3.85	4.06	5.94	7.50	5.10	7.40
209.8	28.5	189	82.82	3.17	4.23	5.82	4.23	9.52	7.94	4.02	5.08	7.09	10.16	10.16	11.01
227.0	24	207	76.03	3.38	3.86	6.28	5.80	3.86	6.28	3.77	4.73	6.86	10.14	5.89	9.66
160.0	18	160	10.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
550.0	24	546	583.65	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.95	1.21	0.73	0.73
428.8	22.5	412	43.17	2.91	2.91	2.91	2.91	2.91	2.91	2.91	2.91	2.91	4.08	2.96	4.08
440.4	22.5	437	24.94	0.00	0.00	0.23	0.00	0.00	0.00	0.00	0.00	0.37	0.09	0.14	0.78
197.6	28.5	184	14.01	0.00	0.00	0.00	0.00	1.09	3.80	0.00	0.00	0.65	1.74	1.96	7.39
111.4	27	99	80.18	2.02	2.02	4.04	2.02	4.04	7.07	2.02	2.02	5.86	3.43	8.69	12.53
111.2	27	93	226.14	0.00	0.00	10.75	0.00	11.83	3.23	0.00	0.65	10.75	4.95	13.98	19.57
189.6	28.5	166	406.18	0.00	0.00	1.81	1.20	3.61	4.82	0.00	0.00	1.93	6.75	4.82	14.22
241.6	21	227	281.43	0.44	0.44	4.41	0.44	1.76	1.76	1.50	0.97	5.02	3.88	2.29	6.43
104.0	19.5	97	9.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.68	0.00	1.24	7.22
128.6	25.5	113	164.50	0.00	0.00	6.19	0.00	6.19	10.62	0.18	3.01	6.90	8.50	8.67	13.81
445.6	25.5	442	249.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.09	0.81
275.4	19.5	272	20.39	0.37	0.37	1.10	0.74	0.37	0.74	0.59	0.37	1.10	0.88	0.37	1.25
321.8	25.5	311	1456.71	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.90	0.64	1.41	3.47
447.2	25.5	442	135.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.36	0.18	1.18
426.6	25.5	409	172.57	0.73	0.73	0.73	0.73	0.73	1.22	0.73	0.73	0.78	0.93	1.32	4.30
536.4	28.5	475	1505.38	1.68	1.68	4.21	6.95	6.74	9.05	2.06	2.36	5.14	10.23	7.96	12.93
733.6	24	697	387.21	0.00	0.00	0.00	2.01	2.01	3.01	0.00	0.00	2.32	3.04	2.21	5.25
238.4	28.5	227	10.15	2.20	2.20	3.52	2.20	2.20	3.52	2.20	2.47	3.70	3.52	4.05	5.02
842.0	21	842	18.52	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
596.0	19.5	590	4.03	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	1.19	1.02	0.85	1.02
1065.0	22.5	996	395.91	3.11	3.11	3.31	3.31	3.11	4.92	3.11	3.13	3.61	8.80	3.15	6.93
1324.8	27	1279	12.42	0.00	0.00	0.00	0.00	0.00	0.78	0.00	0.00	0.00	0.08	0.08	3.58
389.0	28.5	388	5.71	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.05	0.36	0.26
1597.0	27	1408	57.33	0.00	0.00	0.00	0.00	0.00	0.21	0.00	0.00	0.17	0.00	0.31	13.42
729.6	22.5	712	792.86	0.00	0.00	0.98	1.69	0.00	2.39	0.20	0.39	1.54	2.39	0.98	2.47
812.4	21	795	380.25	0.00	0.00	0.00	0.00	0.00	0.88	0.00	0.00	0.81	0.10	0.05	2.19
616.6	28.5	600	63.44	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.43	0.33	0.77	2.77
616.6	19.5	574	68.58	3.31	3.31	3.31	3.31	3.31	5.92	3.31	3.48	4.11	4.25	3.31	7.42
364.6	19.5	311	14.99	7.40	7.40	7.40	7.40	7.40	9.00	7.40	7.40	7.40	7.40	7.40	17.23
191.4	27	189	11.72	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.85	0.00	0.00	1.27
930.8	18	922	20.02	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.50	0.87	0.33	0.95
897.8	22.5	879	967.19	0.68	0.68	0.80	0.80	0.68	0.68	0.68	0.68	0.80	1.02	0.73	2.14
925.6	21	919	271.12	0.11	0.11	0.11	0.11	0.11	0.22	0.11	0.11	0.13	0.17	0.11	0.72
980.0	28.5	948	214.56	0.00	0.00	0.00	0.00	0.63	2.22	0.00	0.00	0.38	0.82	1.14	3.38
411.4	24	402	143.51	0.00	0.00	0.00	1.24	0.50	1.24	0.00	0.25	0.25	1.29	0.80	2.34
				0.9	0.94	1.85	1.33	1.97	2.52	1	1.16	2.35	2.82	2.66	5.47

Table 6
Medium-scale test problems: Best and Average tardiness, ET in seconds, Best/Optimal and Elapse Time of CPLEX, and the Best and the Average RPD values.

						, $m = 2$						
		H + SA		H + VNS		PSO-GA		BKRGA-DE		IG		ABC-TS
Test problem	(n,m)	Best	Average	Best	Average	Best	Average	Best	Average	Best	Average	Best
n2m1P1S1r1-01	(46,2)	435	435.0	435	435.0	467	482.2	435	438.2	446	449.8	451
n2m1P1S1r1-02	(42,2)	70	70.0	70	70.0	70	70.0	70	70.0	70	70.0	70
n2m1P1S1r1-03	(45,2)	372	372.0	372	372.0	409	449.4	372	372.0	372	375.0	372
n2m1P1S1r1-04	(27,2)	234	234.4	235	237.0	248	270.0	235	246.6	241	244.6	243
n2m1P1S1r1-05	(26,2)	213	213.0	213	213.4	229	241.0	213	216.2	213	213.8	221
n2m1P1S1r2-01	(32,2)	548	548.0	550	557.6	619	665.4	554	577.8	589	594.6	620
n2m1P1S1r2-02	(39,2)	1823	1828.6	1830	1832.2	1921	1962.4	1834	1853.8	1850	1882.4	1891
n2m1P1S1r2-03	(35,2)	1354	1354.8	1354	1356.4	1398	1426.8	1356	1367.2	1380	1386.6	1437
n2m1P1S1r2-04	(23,2)	538	538.0	538	538.0	538	541.2	542	550.2	538	540.0	543
n2m1P1S1r2-05	(25,2)	322	322.0	322	322.2	323	328.0	322	324.4	322	322.0	322
n2m1P1S2r1-01	(42,2)	167	167.0	167	167.0	169	198.0	167	169.4	167	168.8	170
n2m1P1S2r1-02	(33,2)	532	532.0	538	544.4	621	633.2	532	551.8	575	582.0	601
n2m1P1S2r1-03	(25,2)	264	264.0	264	264.0	267	274.6	264	265.0	264	266.6	266
n2m1P1S2r1-04	(40,2)	456	458.0	456	462.6	566	567.6	460	472.2	526	564.0	532
n2m1P1S2r1-05	(45,2)	257	257.0	257	257.0	268	277.6	257	257.0	257	261.8	257
n2m1P1S2r2-01	(35,2)	584	585.2	588	591.6	637	657.8	592	604.6	645	661.4	620
n2m1P1S2r2-02	(42,2)	1083	1083.0	1083	1083.0	1124	1172.0	1083	1086.0	1087	1107.8	1085
n2m1P1S2r2-03	(39,2)	867	867.0	867	867.0	981	1001.6	867	875.2	913	959.0	899
n2m1P1S2r2-04	(32,2)	559	559.0	559	559.0	588	613.0	561	564.0	559	566.0	562
n2m1P1S2r2-05	(48,2)	2785	2785.4	2789	2796.0	2892	3044.0	2791	2808.6	2942	3055.6	2888
n2m1P2S1r1-01	(37,2)	590	598.2	598	603.6	637	667.2	598	633.0	616	640.0	600
n2m1P2S1r1-02	(36,2)	402	402.0	416	422.4	448	474.0	455	471.2	438	457.0	465
n2m1P2S1r1-03	(32,2)	768	780.4	796	816.6	908	938.2	826	843.0	873	891.6	873
n2m1P2S1r1-04	(29,2)	951	951.0	951	951.0	951	976.6	951	962.6	951	956.0	951
n2m1P2S1r1-05	(46,2)	2321	2321.0	2321	2322.8	2374	2575.4	2325	2330.6	2327	2347.2	2373
n2m1P2S1r2-01	(46,2)	3679	3679.0	3679	3679.0	3798	3870.8	3679	3683.0	3695	3703.0	3808
n2m1P2S1r2-02	(47,2)	3180	3182.4	3209	3218.8	4591	4722.4	3239	3329.4	3404	3465.0	3478
n2m1P2S1r2-03	(41,2)	4188	4188.6	4188	4190.4	4363	4498.6	4188	4197.6	4200	4223.2	4243
n2m1P2S1r2-04	(23,2)	1012	1012.0	1012	1012.0	1015	1024.4	1012	1014.6	1012	1012.0	1012
n2m1P2S1r2-05	(44,2)	2408	2408.0	2408	2411.2	2503	2556.0	2408	2437.6	2427	2458.6	2440
n2m1P2S2r1-01	(27,2)	643	643.4	643	650.8	711	744.4	645	674.8	664	700.6	672
n2m1P2S2r1-02	(45,2)	504	504.0	504	504.0	556	622.2	504	507.8	540	573.0	504
n2m1P2S2r1-03	(28,2)	629	629.0	629	636.4	710	728.6	645	656.2	655	666.8	709
n2m1P2S2r1-04	(24,2)	324	324.4	325	327.8	358	370.2	333	358.4	345	364.8	370
n2m1P2S2r1-05	(22,2)	631	633.0	631	645.4	646	696.6	644	676.0	642	649.0	664
n2m1P2S2r2-01	(46,2)	4711	4711.8	4712	4720.0	5374	5414.2	4712	4734.0	5008	5179.2	4993
n2m1P2S2r2-02	(29,2)	2232	2232.0	2232	2232.0	2235	2243.6	2232	2232.8	2232	2232.0	2232
n2m1P2S2r2-03	(29,2)	1287	1287.6	1290	1296.0	1342	1353.6	1299	1313.2	1323	1335.4	1343
n2m1P2S2r2-04	(49,2)	3555	3555.0	3555	3559.8	3983	4047.6	3571	3575.6	3720	3900.8	3783
n2m1P2S2r2-05	(44,2)	4035	4035.0	4035	4036.6	4213	4275.8	4035	4043.2	4121	4158.6	4068
Average												

						, $m = 2$									
ABC-TS		CPLEX				RPD_{Best}						$RPD_{Average}$			
Average	ET(sec)	Best/Optimal	Elapsed Time(sec)	H + SA	H + VNS	PSO-GA	BKRGA-DE	IG	ABC-TS	H + SA	H + VNS	PSO-GA	BKRGA-DE	IG	ABC-TS
461.2	82.8	435	1800.00	0.00	0.00	7.36	0.00	2.53	3.68	0.00	0.00	10.85	0.74	3.40	6.02
70.0	75.6	70	231.59	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
372.4	81	372	567.19	0.00	0.00	9.95	0.00	0.00	0.00	0.00	0.00	20.81	0.00	0.81	0.11
250.2	48.6	234	1800.00	0.00	0.43	5.98	0.43	2.99	3.85	0.17	1.28	15.38	5.38	4.53	6.92
229.4	46.8	213	1329.66	0.00	0.00	7.51	0.00	0.00	3.76	0.00	0.19	13.15	1.50	0.38	7.70
638.4	57.6	542	1800.00	1.11	1.48	14.21	2.21	8.67	14.39	1.11	2.88	22.77	6.61	9.70	17.79
1943.2	70.2	1834	1800.00	0.00	0.38	5.38	0.60	1.48	3.73	0.31	0.50	7.65	1.69	3.26	6.59
1492.2	63	1369	1800.00	0.00	0.00	3.25	0.15	1.92	6.13	0.06	0.18	5.38	0.97	2.41	10.21
545.8	41.4	538	189.07	0.00	0.00	0.00	0.74	0.00	0.93	0.00	0.00	0.59	2.27	0.37	1.45
322.0	45	322	109.41	0.00	0.00	0.31	0.00	0.00	0.00	0.00	0.06	1.86	0.75	0.00	0.00
181.6	75.6	167	1800.00	0.00	0.00	1.20	0.00	0.00	1.80	0.00	0.00	18.56	1.44	1.08	8.74
632.2	59.4	530	1800.00	0.38	1.51	17.17	0.38	8.49	13.40	0.38	2.72	19.47	4.11	9.81	19.28
270.2	45	264	1504.48	0.00	0.00	1.14	0.00	0.00	0.76	0.00	0.00	4.02	0.38	0.98	2.35
572.4	72	463	1800.00	0.00	0.00	24.12	0.88	15.35	16.67	0.44	1.45	24.47	3.55	23.68	25.53
265.4	81	257	1800.00	0.00	0.00	4.28	0.00	0.00	0.00	0.00	0.00	8.02	0.00	1.87	3.27
650.2	63	589	1800.00	0.00	0.68	9.08	1.37	10.45	6.16	0.21	1.30	12.64	3.53	13.25	11.34
1134.8	75.6	1090	1800.00	0.00	0.00	3.79	0.00	0.37	0.18	0.00	0.00	8.22	0.28	2.29	4.78
976.4	70.2	921	1800.00	0.00	0.00	13.15	0.00	5.31	3.69	0.00	0.00	15.52	0.95	10.61	12.62
580.4	57.6	559	1800.00	0.00	0.00	5.19	0.36	0.00	0.54	0.00	0.00	9.66	0.89	1.25	3.83
2977.4	86.4	2906	1800.00	0.00	0.14	3.84	0.22	5.64	3.70	0.01	0.39	9.30	0.85	9.72	6.91
640.2	66.6	590	851.84	0.00	1.36	7.97	1.36	4.41	1.69	1.39	2.31	13.08	7.29	8.47	8.51
475.6	64.8	402	1800.00	0.00	3.48	11.44	13.18	8.96	15.67	0.00	5.07	17.91	17.21	13.68	18.31
905.0	57.6	770	1800.00	0.00	3.65	18.23	7.55	13.67	13.67	1.61	6.33	22.16	9.77	16.09	17.84
959.8	52.2	951	971.85	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.69	1.22	0.53	0.93
2441.6	82.8	2344	1800.00	0.00	0.00	2.28	0.17	0.26	2.24	0.00	0.08	10.96	0.41	1.13	5.20
4103.4	82.8	3685	1800.00	0.00	0.00	3.23	0.00	0.43	3.51	0.00	0.00	5.21	0.11	0.65	11.54
3652.0	84.6	3187	1800.00	0.00	0.91	44.37	1.86	7.04	9.37	0.08	1.22	48.50	4.70	8.96	14.84
4458.0	73.8	4200	1800.00	0.00	0.00	4.18	0.00	0.29	1.31	0.01	0.06	7.42	0.23	0.84	6.45
1014.5	41.4	1012	195.23	0.00	0.00	0.30	0.00	0.00	0.00	0.00	0.00	1.23	0.26	0.00	0.25
2478.5	79.2	2420	1800.00	0.00	0.00	3.95	0.00	0.79	1.33	0.00	0.13	6.15	1.23	2.10	2.93
699.8	48.6	643	1800.00	0.00	0.00	10.58	0.31	3.27	4.51	0.06	1.21	15.77	4.95	8.96	8.83
556.3	81	504	1800.00	0.00	0.00	10.32	0.00	7.14	0.00	0.00	0.00	23.45	0.75	13.69	10.37
744.0	50.4	628	1800.00	0.16	0.16	13.06	2.71	4.30	12.90	0.16	1.34	16.02	4.49	6.18	18.47
388.3	43.2	334	1800.00	0.00	0.31	10.49	2.78	6.48	14.20	0.12	1.17	14.26	10.62	12.59	19.83
696.0	39.6	631	1800.00	0.00	0.00	2.38	2.06	1.74	5.23	0.32	2.28	10.40	7.13	2.85	10.30
5266.8	82.8	4734	1800.00	0.00	0.02	14.07	0.02	6.30	5.99	0.02	0.19	14.93	0.49	9.94	11.80
2233.5	52.2	2232	1800.00	0.00	0.00	0.13	0.00	0.00	0.00	0.00	0.00	0.52	0.04	0.00	0.07
1369.8	52.2	1282	1800.00	0.39	0.62	4.68	1.33	3.20	4.76	0.44	1.09	5.59	2.43	4.17	6.84
3882.3	88.2	3571	1800.00	0.00	0.00	12.04	0.45	4.64	6.41	0.00	0.14	13.86	0.58	9.73	9.21
4227.8	79.2	4050	1800.00	0.00	0.00	4.41	0.00	2.13	0.82	0.00	0.04	5.97	0.20	3.06	4.78
				0.05	0.38	7.87	1.03	3.46	4.67	0.17	0.84	12.11	2.75	5.58	8.57

Table 7
Medium-scale test problems: Best and Average tardiness, ET in seconds, Best/Optimal and Elapse Time of CPLEX, and the Best and the Average RPD values.

		, m=4										
		H + SA		H + VNS		PSO-GA		BKRGA-DE		IG		ABC-TS
Test problem	(n,m)	Best	Average	Best	Average	Best	Average	Best	Average	Best	Average	Best
n2m2P1S1r1-01	(29,4)	146	146.0	146	146.0	155	163.8	146	147.2	146	146.0	146
n2m2P1S1r1-02	(48,4)	375	375.0	375	375.0	375	375.0	375	375.0	375	375.0	375
n2m2P1S1r1-03	(21,4)	115	115.0	115	115.0	117	119.0	115	116.2	115	115.2	116
n2m2P1S1r1-04	(37,4)	239	239.0	239	239.0	239	242.0	239	239.0	239	239.0	239
n2m2P1S1r1-05	(36,4)	226	226.0	226	226.0	226	226.0	226	226.0	226	226.0	226
n2m2P1S1r2-01	(44,4)	1395	1395.0	1395	1395.0	1417	1446.8	1395	1395.6	1395	1395.0	1397
n2m2P1S1r2-02	(40,4)	1184	1184.0	1184	1184.0	1197	1229.3	1184	1186.0	1184	1184.0	1187
n2m2P1S1r2-03	(26,4)	674	675.6	676	676.0	691	709.8	676	676.0	674	676.0	694
n2m2P1S1r2-04	(38,4)	997	997.0	997	997.0	997	1000.6	997	997.0	997	997.0	997
n2m2P1S1r2-05	(38,4)	780	780.0	780	780.0	784	786.6	780	780.0	780	780.0	780
n2m2P1S2r1-01	(28,4)	146	146.0	146	146.0	146	146.0	146	146.0	146	146.0	146
n2m2P1S2r1-02	(25,4)	59	59.0	59	59.0	59	59.2	59	60.2	59	59.0	59
n2m2P1S2r1-03	(38,4)	226	226.0	226	226.0	226	226.0	226	226.0	226	226.0	226
n2m2P1S2r1-04	(21,4)	66	66.4	66	66.6	67	69.2	67	67.4	67	67.0	67
n2m2P1S2r1-05	(40,4)	340	340.0	340	340.0	340	343.0	340	340.0	340	340.0	340
n2m2P1S2r2-01	(31,4)	473	473.0	473	473.0	473	473.6	473	473.0	473	473.0	473
n2m2P1S2r2-02	(27,4)	505	505.0	505	505.0	505	505.0	505	505.0	505	505.0	505
n2m2P1S2r2-03	(42,4)	1037	1037.0	1037	1037.0	1230	1290.4	1037	1230.8	1037	1042.0	1341
n2m2P1S2r2-04	(35,4)	668	668.0	668	668.0	668	669.0	668	668.0	668	668.0	668
n2m2P1S2r2-05	(41,4)	2273	2273.0	2273	2273.0	2273	2290.0	2273	2273.4	2273	2273.0	2273
n2m2P2S1r1-01	(36,4)	336	336.0	336	336.0	336	354.0	336	336.0	336	336.0	336
n2m2P2S1r1-02	(34,4)	412	412.0	412	412.0	412	412.0	412	412.0	412	412.0	412
n2m2P2S1r1-03	(49,4)	1013	1013.0	1013	1013.0	1019	1074.0	1013	1013.0	1013	1013.0	1013
n2m2P2S1r1-04	(40,4)	734	734.0	738	743.0	822	863.4	740	753.2	752	756.6	786
n2m2P2S1r1-05	(40,4)	546	546.0	546	546.0	546	560.8	546	546.0	546	546.0	546
n2m2P2S1r2-01	(42,4)	5447	5447.0	5447	5447.0	5513	5606.2	5447	5447.0	5447	5447.0	5452
n2m2P2S1r2-02	(42,4)	2929	2929.0	2929	2929.0	2996	3232.2	2929	2929.0	2929	2929.0	2929
n2m2P2S1r2-03	(28,4)	576	576.0	576	576.0	586	594.8	576	576.0	576	576.0	622
n2m2P2S1r2-04	(32,4)	2345	2345.0	2345	2345.0	2370	2394.8	2345	2347.0	2345	2345.0	2356
n2m2P2S1r2-05	(44,4)	3112	3112.0	3112	3112.0	3373	3554.2	3810	3834.7	3112	3112.0	3645
n2m2P2S2r1-01	(37,4)	550	550.0	550	550.0	550	559.8	550	553.6	550	550.6	550
n2m2P2S2r1-02	(33,4)	436	436.0	436	436.0	436	450.2	436	436.0	436	436.0	436
n2m2P2S2r1-03	(32,4)	667	667.0	667	667.2	667	671.8	667	667.6	667	667.0	667
n2m2P2S2r1-04	(36,4)	331	331.0	331	331.0	334	338.4	331	331.0	331	331.0	331
n2m2P2S2r1-05	(37,4)	451	451.0	451	451.0	451	451.0	451	451.0	451	451.0	451
n2m2P2S2r2-01	(24,4)	769	769.0	769	772.4	769	806.4	769	786.0	769	769.0	796
n2m2P2S2r2-02	(24,4)	815	815.0	815	815.0	815	815.0	815	815.0	815	815.0	815
n2m2P2S2r2-03	(44,4)	3187	3187.0	3187	3187.0	3331	3462.8	3187	3484.6	3187	3187.0	3191
n2m2P2S2r2-04	(38,4)	2851	2851.0	2851	2851.0	2880	2920.2	2851	2851.0	2851	2853.6	3473
n2m2P2S2r2-05	(38,4)	3660	3660.0	3660	3660.0	3708	3775.2	3660	3710.2	3660	3660.0	3672
Average												

, m=4															
ABC-TS		CPLEX		{RPD\vphantom{RPD}_{{Best\vphantom{Best}}}}						{RPD\vphantom{RPD}_{{Average\vphantom{Average}}}}					
Average	ET(sec)	Best/Optimal	Elapsed Time(sec)	H + SA	H + VNS	PSO-GA	BKRGA-DE	IG	ABC-TS	H + SA	H + VNS	PSO-GA	BKRGA-DE	IG	ABC-TS
146.0	52.2	146	29.17	0.00	0.00	6.16	0.00	0.00	0.00	0.00	0.00	12.19	0.82	0.00	0.00
375.0	86.4	375	339.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
121.6	37.8	115	0.93	0.00	0.00	1.74	0.00	0.00	0.87	0.00	0.00	3.48	1.04	0.17	5.74
239.0	66.6	239	34.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.26	0.00	0.00	0.00
226.0	64.8	226	96.53	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1453.6	79.2	1395	1800.00	0.00	0.00	1.58	0.00	0.00	0.14	0.00	0.00	3.71	0.04	0.00	4.20
1192.6	72	1184	1800.00	0.00	0.00	1.10	0.00	0.00	0.25	0.00	0.00	3.83	0.17	0.00	0.73
703.3	46.8	735	1800.00	0.00	0.30	2.52	0.30	0.00	2.97	0.24	0.30	5.31	0.30	0.30	4.34
997.0	68.4	997	123.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.36	0.00	0.00	0.00
780.0	68.4	780	1800.00	0.00	0.00	0.51	0.00	0.00	0.00	0.00	0.00	0.85	0.00	0.00	0.00
146.0	50.4	146	295.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
60.6	45	59	476.94	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.34	2.03	0.00	2.71
226.0	68.4	226	1684.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
68.8	37.8	66	249.95	0.00	0.00	1.52	1.52	1.52	1.52	0.61	0.91	4.85	2.12	1.52	4.24
340.6	72	340	640.84	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.88	0.00	0.00	0.18
473.0	55.8	473	1800.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.00	0.00	0.00
505.0	48.6	505	70.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1431.0	75.6	1037	586.71	0.00	0.00	18.61	0.00	0.00	29.32	0.00	0.00	24.44	18.69	0.48	37.99
668.0	63	668	117.93	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.15	0.00	0.00	0.00
2274.6	73.8	2273	1800.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.02	0.00	0.07
336.0	64.8	336	510.58	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.36	0.00	0.00	0.00
412.0	61.2	412	164.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1013.0	88.2	1013	248.62	0.00	0.00	0.59	0.00	0.00	0.00	0.00	0.00	6.02	0.00	0.00	0.00
796.6	72	734	1800.00	0.00	0.54	11.99	0.82	2.45	7.08	0.00	1.23	17.63	2.62	3.08	8.53
546.0	72	546	323.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.71	0.00	0.00	0.00
5503.2	75.6	5447	308.83	0.00	0.00	1.21	0.00	0.00	0.09	0.00	0.00	2.92	0.00	0.00	1.03
2938.4	75.6	2929	1800.00	0.00	0.00	2.29	0.00	0.00	0.00	0.00	0.00	10.35	0.00	0.00	0.32
635.0	50.4	575	51.78	0.17	0.17	1.91	0.17	0.17	8.17	0.17	0.17	3.44	0.17	0.17	10.43
2362.0	57.6	2345	280.82	0.00	0.00	1.07	0.00	0.00	0.47	0.00	0.00	2.12	0.09	0.00	0.72
3657.5	79.2	3112	1800.00	0.00	0.00	8.39	22.43	0.00	17.13	0.00	0.00	14.21	23.22	0.00	17.53
552.8	66.6	550	1800.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.78	0.65	0.11	0.51
436.0	59.4	436	100.81	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.26	0.00	0.00	0.00
667.4	57.6	667	122.66	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.72	0.09	0.00	0.06
332.4	64.8	331	334.51	0.00	0.00	0.91	0.00	0.00	0.00	0.00	0.00	2.24	0.00	0.00	0.42
451.0	66.6	451	296.59	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
812.4	43.2	769	56.13	0.00	0.00	0.00	0.00	0.00	3.51	0.00	0.44	4.86	2.21	0.00	5.64
815.0	43.2	815	59.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3555.0	79.2	3187	1800.00	0.00	0.00	4.52	0.00	0.00	0.13	0.00	0.00	8.65	9.34	0.00	11.55
3697.0	68.4	2851	1800.00	0.00	0.00	1.02	0.00	0.00	21.82	0.00	0.00	2.43	0.00	0.09	29.67
3715.5	68.4	3660	310.12	0.00	0.00	1.31	0.00	0.00	0.33	0.00	0.00	3.15	1.37	0.00	1.52
				0.00	0.03	1.72	0.63	0.1	2.34	0.03	0.08	3.86	1.62	0.15	3.7

Table 8

Large-scale test problems: Best and Average tardiness, ET in seconds, Best of All, and the Best and the Average RPD values.

, m=2}

		H + SA		H + VNS		PSO-GA		BKRGA-DE		IG		ABC-TS
Test problem	(n,m)	Best	Average	Best	Average	Best	Average	Best	Average	Best	Average	Best
n3m1P1S1r1-01	(57,2)	445	445.0	445	445.0	483	502.6	445	447.4	445	445.0	445
n3m1P1S1r1-02	(72,2)	1114	1114.0	1114	1117.8	1333	1378.6	1128	1165.6	1168	1168.4	1213
n3m1P1S1r1-03	(64,2)	452	452.0	452	452.8	555	575.6	452	477.0	465	465.4	452
n3m1P1S1r1-04	(83,2)	1034	1034.0	1034	1046.8	1349	1389.8	1058	1202.6	1087	1086.6	1069
n3m1P1S1r1-05	(64,2)	223	223.0	223	225.4	241	254.2	223	227.2	225	225.4	223
n3m1P1S1r2-01	(71,2)	3732	3732.0	3735	3753.6	4365	4732.2	3897	4157.4	3954	3953.6	4027
n3m1P1S1r2-02	(60,2)	3617	3617.0	3634	3644.6	3945	4494.6	3647	3771.0	3693	3693.0	4370
n3m1P1S1r2-03	(82,2)	4415	4415.0	4419	4452.4	5506	5946.2	4606	5326.4	4634	4634.4	5389
n3m1P1S1r2-04	(71,2)	3413	3413.5	3413	3415.8	4105	4652.4	3436	3761.2	3500	3500.2	3547
n3m1P1S1r2-05	(96,2)	7885	7887.5	7952	8156.6	9605	10906.4	8573	9373.0	9797	9797.0	10312
n3m1P1S2r1-01	(87,2)	782	782.0	782	785.0	929	1138.6	789	837.8	868	868.0	875
n3m1P1S2r1-02	(84,2)	2016	2016.0	2027	2051.0	2234	2528.6	2041	2238.6	2435	2434.8	2430
n3m1P1S2r1-03	(78,2)	616	616.0	616	620.2	727	919.6	620	670.6	685	684.6	794
n3m1P1S2r1-04	(68,2)	821	821.0	850	859.6	955	1056.6	851	978.0	1012	1012.4	997
n3m1P1S2r1-05	(76,2)	936	936.0	936	938.4	1071	1174.8	938	960.6	1036	1035.6	1036
n3m1P1S2r2-01	(51,2)	2996	2996.0	2996	2998.8	3159	3262.4	3007	3037.4	3107	3106.8	3169
n3m1P1S2r2-02	(71,2)	4220	4220.0	4247	4285.0	4881	5233.0	4393	4626.4	4969	4969.4	5060
n3m1P1S2r2-03	(80,2)	4222	4222.0	4251	4264.4	5008	5497.8	4658	4907.4	4985	4984.8	5638
n3m1P1S2r2-04	(68,2)	3382	3382.0	3388	3403.2	4147	4458.6	3502	3796.2	3715	3714.6	4171
n3m1P1S2r2-05	(86,2)	4236	4236.0	4243	4314.6	5047	5433.2	4488	4940.4	5303	5302.6	6795
n3m1P2S1r1-01	(68,2)	2032	2032.0	2032	2038.4	2142	2394.6	2040	2100.8	2055	2055.4	2089
n3m1P2S1r1-02	(71,2)	1971	1971.0	1971	1980.0	2290	2384.8	2006	2117.6	1995	1995.4	1983
n3m1P2S1r1-03	(68,2)	1172	1172.5	1172	1172.0	1366	1431.6	1174	1252.0	1174	1173.6	1248
n3m1P2S1r1-04	(94,2)	3596	3596.0	3596	3605.0	4631	4739.2	4223	5072.2	3718	3718.0	4474
n3m1P2S1r1-05	(63,2)	2401	2401.0	2413	2429.2	2698	2854.0	2476	2509.0	2504	2503.6	2447
n3m1P2S1r2-01	(69,2)	7691	7691.5	7701	7758.4	8365	8665.8	7994	8372.4	8122	8122.4	8565
n3m1P2S1r2-02	(77,2)	12021	12022.0	12041	12061.2	13718	13988.4	13250	14157.6	12457	12457.0	13057
n3m1P2S1r2-03	(64,2)	6459	6459.0	6495	6501.8	7403	7626.5	6562	6862.0	6687	6687.4	6878
n3m1P2S1r2-04	(51,2)	5839	5839.0	5847	5869.0	6786	7212.5	5883	5946.8	5954	5954.0	5999
n3m1P2S1r2-05	(88,2)	18758	18766.0	18953	19171.4	24159	25445.0	22376	24723.6	21945	21945.0	24399
n3m1P2S2r1-01	(86,2)	1196	1196.0	1196	1196.4	1251	1339.8	1199	1298.4	1221	1221.0	1368
n3m1P2S2r1-02	(61,2)	1220	1220.0	1220	1220.0	1473	1573.8	1220	1230.2	1235	1234.6	1262
n3m1P2S2r1-03	(66,2)	1199	1199.0	1199	1206.8	1285	1353.0	1209	1249.4	1240	1240.2	1313
n3m1P2S2r1-04	(83,2)	1672	1672.0	1673	1680.8	1927	2051.8	1837	2078.6	2032	2032.4	1806
n3m1P2S2r1-05	(79,2)	3898	3898.0	3914	3929.6	4035	4272.3	3936	4006.4	4503	4503.4	4208
n3m1P2S2r2-01	(87,2)	8917	8960.5	8997	9052.6	10768	11117.5	9777	11188.2	10629	10628.6	11153
n3m1P2S2r2-02	(73,2)	6873	6873.0	6878	6909.8	7476	7590.8	6983	7444.2	7857	7857.2	7654
n3m1P2S2r2-03	(60,2)	6006	6008.5	6006	6023.4	6447	6527.0	6347	6390.0	6396	6396.4	7100
n3m1P2S2r2-04	(90,2)	14332	14337.0	14422	14512.8	17875	18184.0	15798	17690.2	17778	17778.4	19782
n3m1P2S2r2-05	(56,2)	5657	5657.5	5671	5687.0	6415	6644.3	5728	5795.6	6022	6022.2	6309
Average												

, m=2}

ABC-TS			RPD_{Best}						$RPD_{Average}$					
Average	ET(sec)	Best of All	H + SA	H + VNS	PSO-GA	BKRGA-DE	IG	ABC-TS	H + SA	H + VNS	PSO-GA	BKRGA-DE	IG	ABC-TS
454.6	102.6	445	0.00	0.00	8.54	0.00	0.00	0.00	0.00	0.00	12.94	0.54	0.00	2.16
1291.4	129.6	1114	0.00	0.00	19.66	1.26	4.88	8.89	0.00	0.34	23.75	4.63	4.88	15.92
471.6	115.2	452	0.00	0.00	22.79	0.00	2.96	0.00	0.00	0.18	27.35	5.53	2.96	4.34
1106.8	149.4	1034	0.00	0.00	30.46	2.32	5.09	3.38	0.00	1.24	34.41	16.31	5.09	7.04
227.2	115.2	223	0.00	0.00	8.07	0.00	1.08	0.00	0.00	1.08	13.99	1.88	1.08	1.88
4275.8	127.8	3732	0.00	0.08	16.96	4.42	5.94	7.90	0.00	0.58	26.80	11.40	5.94	14.57
4556.0	108	3617	0.00	0.47	9.07	0.83	2.10	20.82	0.00	0.76	24.26	4.26	2.10	25.96
5822.6	147.6	4415	0.00	0.09	24.71	4.33	4.97	22.06	0.00	0.85	34.68	20.64	4.97	31.88
3684.8	127.8	3413	0.00	0.00	20.28	0.67	2.55	3.93	0.01	0.08	36.31	10.20	2.55	7.96
11127.6	172.8	7885	0.00	0.85	21.81	8.73	24.25	30.78	0.03	3.44	38.32	18.87	24.25	41.12
1031.2	156.6	782	0.00	0.00	18.80	0.90	11.00	11.89	0.00	0.38	45.60	7.14	11.00	31.87
2566.6	151.2	2016	0.00	0.55	10.81	1.24	20.77	20.54	0.00	1.74	25.43	11.04	20.77	27.31
918.2	140.4	616	0.00	0.00	18.02	0.65	11.14	28.90	0.00	0.68	49.29	8.86	11.14	49.06
1068.8	122.4	821	0.00	3.53	16.32	3.65	23.31	21.44	0.00	4.70	28.70	19.12	23.31	30.18
1220.2	136.8	936	0.00	0.00	14.42	0.21	10.64	10.68	0.00	0.26	25.51	2.63	10.64	30.36
3215.0	91.8	2996	0.00	0.00	5.44	0.37	3.70	5.77	0.00	0.09	8.89	1.38	3.70	7.31
5257.0	127.8	4220	0.00	0.64	15.66	4.10	17.76	19.91	0.00	1.54	24.00	9.63	17.76	24.57
5961.0	144	4222	0.00	0.69	18.62	10.33	18.07	33.54	0.00	1.00	30.22	16.23	18.07	41.19
4363.8	122.4	3382	0.00	0.18	22.62	3.55	9.83	23.33	0.00	0.63	31.83	12.25	9.83	29.03
7149.2	154.8	4236	0.00	0.17	19.15	5.95	25.18	60.41	0.00	1.86	28.26	16.63	25.18	68.77
2138.4	122.4	2032	0.00	0.00	5.41	0.39	1.15	2.81	0.00	0.31	17.84	3.39	1.15	5.24
2056.4	127.8	1971	0.00	0.00	16.18	1.78	1.24	0.61	0.00	0.46	20.99	7.44	1.24	4.33
1319.8	122.4	1172	0.00	0.00	16.55	0.17	0.14	6.48	0.04	0.00	22.15	6.83	0.14	12.61
5938.8	169.2	3596	0.00	0.00	28.78	17.44	3.39	24.42	0.00	0.25	31.79	41.05	3.39	65.15
2510.4	113.4	2401	0.00	0.50	12.37	3.12	4.27	1.92	0.00	1.17	18.87	4.50	4.27	4.56
9272.4	124.2	7691	0.00	0.13	8.76	3.94	5.61	11.36	0.01	0.88	12.67	8.86	5.61	20.56
13443.6	138.6	12021	0.00	0.17	14.12	10.22	3.63	8.62	0.01	0.33	16.37	17.77	3.63	11.83
7664.8	115.2	6459	0.00	0.56	14.62	1.59	3.54	6.49	0.00	0.66	18.08	6.24	3.54	18.67
6268.4	91.8	5839	0.00	0.14	16.22	0.75	1.97	2.74	0.00	0.51	23.52	1.85	1.97	7.35
25106.2	158.4	18758	0.00	1.04	28.79	19.29	16.99	30.07	0.04	2.20	35.65	31.80	16.99	33.84
1703.0	154.8	1196	0.00	0.00	4.60	0.25	2.09	14.38	0.00	0.03	12.02	8.56	2.09	42.39
1381.8	109.8	1220	0.00	0.00	20.74	0.00	1.20	3.44	0.00	0.00	29.00	0.84	1.20	13.26
1374.2	118.8	1199	0.00	0.00	7.17	0.83	3.44	9.51	0.00	0.65	12.84	4.20	3.44	14.61
2149.4	149.4	1672	0.00	0.06	15.25	9.87	21.56	8.01	0.00	0.53	22.71	24.32	21.56	28.55
4689.0	142.2	3898	0.00	0.41	3.51	0.97	15.53	7.95	0.00	0.81	9.60	2.78	15.53	20.29
12509.8	156.6	8917	0.00	0.90	20.76	9.64	19.19	25.08	0.49	1.52	24.68	25.47	19.19	40.29
8324.0	131.4	6873	0.00	0.07	8.77	1.60	14.32	11.36	0.00	0.54	10.44	8.31	14.32	21.11
7505.6	108	6006	0.00	0.00	7.34	5.68	6.50	18.22	0.04	0.29	8.67	6.39	6.50	24.97
22264.2	162	14332	0.00	0.63	24.72	10.23	24.05	38.03	0.03	1.26	26.88	23.43	24.05	55.35
6589.8	100.8	5657	0.00	0.25	13.40	1.26	6.46	11.53	0.01	0.53	17.45	2.45	6.46	16.49
			0.00	0.3	15.76	3.81	9.04	14.43	0.02	0.86	24.07	10.89	9.04	23.85

Table 9
Large-scale test problems: Best and Average tardiness, ET in seconds, Best of All, and the Best and the Average RPD values.

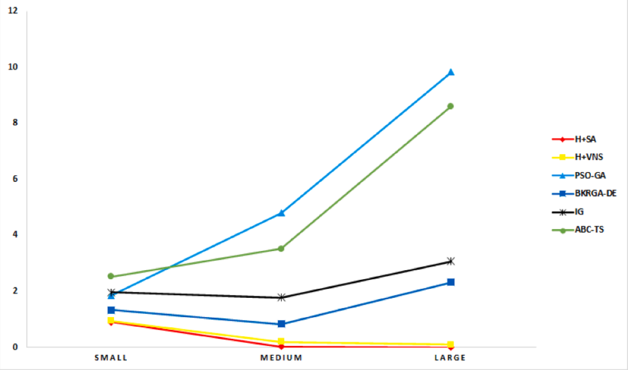
						, m=4						
		H + SA		H + VNS		PSO-GA		BKRGA-DE		IG		ABC-TS
Test problem	(n,m)	Best	Average	Best	Average	Best	Average	Best	Average	Best	Average	Best
n3m2P1S1r1-01	(55,4)	241	241.0	241	241.0	241	241.0	241	241.0	241	241.0	241
n3m2P1S1r1-02	(90,4)	1450	1450.0	1450	1450.0	1659	1931.0	1450	1450.4	1450	1450.0	1450
n3m2P1S1r1-03	(65,4)	550	550.0	550	551.4	594	667.4	550	558.0	550	550.4	551
n3m2P1S1r1-04	(78,4)	468	468.0	468	468.0	530	638.0	468	488.8	468	468.0	606
n3m2P1S1r1-05	(73,4)	526	526.0	526	526.0	631	696.0	526	565.2	526	526.0	530
n3m2P1S1r2-01	(75,4)	3575	3575.0	3575	3575.0	3675	3741.6	3575	3577.8	3575	3575.0	3575
n3m2P1S1r2-02	(80,4)	4355	4355.0	4355	4357.0	5303	5887.6	4395	4941.5	4365	4364.8	5509
n3m2P1S1r2-03	(94,4)	6707	6707.0	6707	6719.2	10265	10283.0	8978	10265.4	6808	6808.2	8182
n3m2P1S1r2-04	(81,4)	3509	3509.0	3509	3510.3	4391	5160.3	3512	5292.0	3510	3509.6	4747
n3m2P1S1r2-05	(73,4)	2450	2450.0	2450	2451.0	2522	3493.6	2450	2450.0	2450	2450.0	2522
n3m2P1S2r1-01	(86,4)	1360	1360.0	1360	1360.0	1404	1503.4	1360	1367.2	1360	1360.0	1360
n3m2P1S2r1-02	(99,4)	1052	1052.0	1052	1052.0	1177	1417.8	1052	1092.4	1052	1052.0	1052
n3m2P1S2r1-03	(59,4)	776	776.0	776	776.0	776	776.2	776	776.0	776	776.0	776
n3m2P1S2r1-04	(56,4)	713	713.0	713	713.0	713	723.4	713	713.0	713	713.0	713
n3m2P1S2r1-05	(86,4)	975	975.0	975	975.0	1015	1100.4	975	987.2	975	975.0	975
n3m2P1S2r2-01	(53,4)	1146	1146.0	1146	1146.0	1146	1171.4	1146	1147.8	1146	1146.0	1146
n3m2P1S2r2-02	(70,4)	2517	2517.0	2517	2517.0	2558	2582.6	2517	2517.4	2517	2517.0	2517
n3m2P1S2r2-03	(67,4)	1252	1252.0	1252	1252.3	1309	1770.6	1252	1293.2	1252	1252.0	1738
n3m2P1S2r2-04	(64,4)	3341	3341.0	3341	3341.0	4560	4590.0	3350	4631.0	3341	3341.0	3954
n3m2P1S2r2-05	(56,4)	1745	1745.0	1745	1745.7	1863	2421.2	1745	2369.0	1745	1745.0	1786
n3m2P2S1r1-01	(57,4)	1311	1311.0	1311	1311.0	1312	1357.8	1311	1311.4	1311	1311.0	1311
n3m2P2S1r1-02	(78,4)	1846	1846.0	1846	1846.0	1848	1932.4	1846	1846.4	1846	1846.0	1846
n3m2P2S1r1-03	(72,4)	1791	1791.0	1791	1791.0	1791	1856.2	1791	1791.4	1791	1791.0	1791
n3m2P2S1r1-04	(98,4)	4291	4291.0	4291	4292.3	4512	5382.2	4347	4430.2	4292	4291.6	4449
n3m2P2S1r1-05	(86,4)	3826	3826.0	3826	3832.7	4974	5450.6	3826	3857.8	3827	3827.4	3876
n3m2P2S1r2-01	(99,4)	16209	16209.0	16209	16236.0	19602	21212.2	17118	18746.8	16334	16334.0	22139
n3m2P2S1r2-02	(70,4)	11095	11095.0	11095	11101.8	11358	11977.0	11095	11108.6	11095	11095.0	11095
n3m2P2S1r2-03	(63,4)	6329	6329.0	6329	6329.0	6587	7172.0	6329	6330.6	6329	6329.0	6412
n3m2P2S1r2-04	(88,4)	17514	17514.0	17514	17555.0	23181	23562.3	18380	23944.8	17670	17670.2	26107
n3m2P2S1r2-05	(87,4)	17104	17104.0	17104	17132.0	22677	23653.0	26581	31629.0	17621	17621.4	30601
n3m2P2S2r1-01	(62,4)	892	892.0	892	892.0	901	941.0	892	892.0	892	892.0	892
n3m2P2S2r1-02	(74,4)	1610	1610.0	1610	1610.0	1625	1703.3	1610	1613.2	1610	1610.0	1610
n3m2P2S2r1-03	(81,4)	2842	2842.0	2842	2844.3	3145	3587.5	2842	2857.8	2842	2842.0	2842
n3m2P2S2r1-04	(53,4)	673	673.0	673	673.0	673	679.5	673	673.8	673	673.0	673
n3m2P2S2r1-05	(88,4)	1117	1117.0	1117	1117.0	1117	1281.6	1117	1117.0	1117	1117.0	1117
n3m2P2S2r2-01	(58,4)	6637	6637.0	6637	6637.0	6637	6639.7	6637	6637.0	6637	6637.0	6637
n3m2P2S2r2-02	(82,4)	10093	10093.0	10093	10093.0	12243	13157.0	10093	10106.6	10093	10093.0	10093
n3m2P2S2r2-03	(90,4)	11496	11496.0	11496	11498.0	11800	13477.0	11509	11634.8	11546	11546.0	11655
n3m2P2S2r2-04	(77,4)	9242	9242.0	9242	9242.0	9488	11024.0	9242	9447.6	9253	9253.4	9437
n3m2P2S2r2-05	(69,4)	8617	8617.0	8617	8617.0	9013	9303.5	8617	8617.6	8617	8617.0	8617
Average												

							, m=4							
ABC-TS			{RPD\vphantom{RPD}_{{Best\vphantom{Best}}}}						{RPD\vphantom{RPD}_{{Average\vphantom{Average}}}}					
Average	ET(sec)	Best of All	H + SA	H + VNS	PSO-GA	BKRGA-DE	IG	ABC-TS	H + SA	H + VNS	PSO-GA	BKRGA-DE	IG	ABC-TS
241.0	99	241	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1450.0	162	1450	0.00	0.00	14.41	0.00	0.00	0.00	0.00	0.00	33.17	0.03	0.00	0.00
557.6	117	550	0.00	0.00	8.00	0.00	0.07	0.18	0.00	0.25	21.35	1.45	0.07	1.38
657.2	140.4	468	0.00	0.00	13.25	0.00	0.00	29.49	0.00	0.00	36.32	4.44	0.00	40.43
574.4	131.4	526	0.00	0.00	19.96	0.00	0.00	0.76	0.00	0.00	32.32	7.45	0.00	9.20
3767.0	135	3575	0.00	0.00	2.80	0.00	0.00	0.00	0.00	0.00	4.66	0.08	0.00	5.37
5894.8	144	4355	0.00	0.00	21.77	0.92	0.23	26.50	0.00	0.05	35.19	13.47	0.23	35.36
9383.4	169.2	6707	0.00	0.00	53.05	33.86	1.51	21.99	0.00	0.18	53.32	53.06	1.51	39.90
4993.0	145.8	3509	0.00	0.00	25.14	0.09	0.02	35.28	0.00	0.04	47.06	50.81	0.02	42.29
2655.4	131.4	2450	0.00	0.00	2.94	0.00	0.00	2.94	0.00	0.04	42.60	0.00	0.00	8.38
1365.6	154.8	1360	0.00	0.00	3.24	0.00	0.00	0.00	0.00	0.00	10.54	0.53	0.00	0.41
1054.0	178.2	1052	0.00	0.00	11.88	0.00	0.00	0.00	0.00	0.00	34.77	3.84	0.00	0.19
776.0	106.2	776	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00
713.0	100.8	713	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.46	0.00	0.00	0.00
975.0	154.8	975	0.00	0.00	4.10	0.00	0.00	0.00	0.00	0.00	12.86	1.25	0.00	0.00
1146.2	95.4	1146	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.22	0.16	0.00	0.02
2520.2	126	2517	0.00	0.00	1.63	0.00	0.00	0.00	0.00	0.00	2.61	0.02	0.00	0.13
1999.4	120.6	1252	0.00	0.00	4.55	0.00	0.00	38.82	0.00	0.03	41.42	3.29	0.00	59.70
4523.2	115.2	3341	0.00	0.00	36.49	0.27	0.00	18.35	0.00	0.00	37.38	38.61	0.00	35.38
2090.2	100.8	1745	0.00	0.00	6.76	0.00	0.00	2.35	0.00	0.04	38.75	35.76	0.00	19.78
1314.0	102.6	1311	0.00	0.00	0.08	0.00	0.00	0.00	0.00	0.00	3.57	0.03	0.00	0.23
1846.0	140.4	1846	0.00	0.00	0.11	0.00	0.00	0.00	0.00	0.00	4.68	0.02	0.00	0.00
1791.0	129.6	1791	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.64	0.02	0.00	0.00
4848.0	176.4	4291	0.00	0.00	5.15	1.31	0.01	3.68	0.00	0.03	25.43	3.24	0.01	12.98
3901.0	154.8	3826	0.00	0.00	30.01	0.00	0.04	1.31	0.00	0.17	42.46	0.83	0.04	1.96
24956.4	178.2	16209	0.00	0.00	20.93	5.61	0.77	36.58	0.00	0.17	30.87	15.66	0.77	53.97
11123.2	126	11095	0.00	0.00	2.37	0.00	0.00	0.00	0.00	0.06	7.95	0.12	0.00	0.25
6655.8	113.4	6329	0.00	0.00	4.08	0.00	0.00	1.31	0.00	0.00	13.32	0.03	0.00	5.16
28185.7	158.4	17514	0.00	0.00	32.36	4.94	0.89	49.06	0.00	0.23	34.53	36.72	0.89	60.93
34178.5	156.6	17104	0.00	0.00	32.58	55.41	3.03	78.91	0.00	0.16	38.29	84.92	3.03	99.83
892.0	111.6	892	0.00	0.00	1.01	0.00	0.00	0.00	0.00	0.00	5.49	0.00	0.00	0.00
1610.0	133.2	1610	0.00	0.00	0.93	0.00	0.00	0.00	0.00	0.00	5.79	0.20	0.00	0.00
2845.8	145.8	2842	0.00	0.00	10.66	0.00	0.00	0.00	0.00	0.08	26.23	0.56	0.00	0.13
673.0	95.4	673	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.97	0.12	0.00	0.00
1129.6	158.4	1117	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	14.74	0.00	0.00	1.13
6638.6	104.4	6637	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.02
10149.2	147.6	10093	0.00	0.00	21.30	0.00	0.00	0.00	0.00	0.00	30.36	0.13	0.00	0.56
11912.4	162	11496	0.00	0.00	2.64	0.11	0.43	1.38	0.00	0.02	17.23	1.21	0.43	3.62
9651.0	138.6	9242	0.00	0.00	2.66	0.00	0.12	2.11	0.00	0.00	19.28	2.22	0.12	4.43
8621.4	124.2	8617	0.00	0.00	4.60	0.00	0.00	0.00	0.00	0.00	7.97	0.01	0.00	0.05
			0.00	0.00	10.04	2.56	0.18	8.78	0.00	0.04	20.53	9.01	0.18	13.58

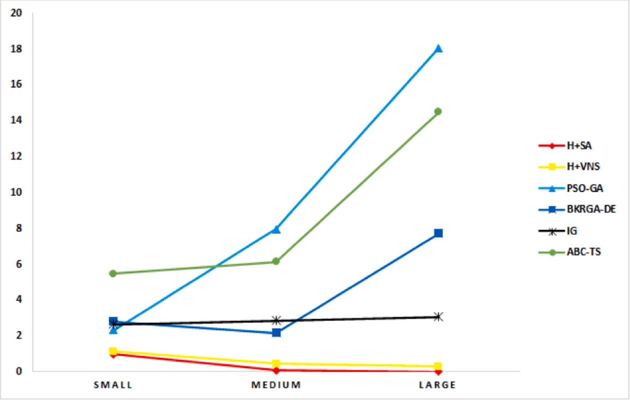
Table 10
Large-scale test problems:Best and Average tardiness, ET in seconds, Best of All, and the Best and the Average RPD values.

						, m=6						
		H + SA		H + VNS		PSO-GA		BKRGA-DE		IG		ABC-TS
Test problem	(n,m)	Best	Average	Best	Average	Best	Average	Best	Average	Best	Average	Best
n3m3P1S1r1-01	(69,6)	603	603.0	603	603.0	603	643.8	603	618.4	603	603.0	603
n3m3P1S1r1-02	(59,6)	339	339.0	339	339.0	339	339.0	339	339.0	339	339.0	339
n3m3P1S1r1-03	(51,6)	165	165.0	165	165.3	165	169.8	165	165.0	165	165.0	165
n3m3P1S1r1-04	(70,6)	677	677.0	677	677.0	677	684.8	677	677.0	677	677.0	677
n3m3P1S1r1-05	(58,6)	627	627.0	627	627.0	638	680.0	627	645.6	627	627.0	645
n3m3P1S1r2-01	(58,6)	1913	1913.0	1913	1913.0	1913	1919.4	1913	1915.7	1913	1913.0	1913
n3m3P1S1r2-02	(59,6)	1931	1931.0	1931	1931.0	1931	1934.4	1931	2077.4	1931	1931.0	1931
n3m3P1S1r2-03	(84,6)	4075	4075.0	4075	4075.0	4463	4501.0	4463	4496.6	4075	4075.0	4484
n3m3P1S1r2-04	(55,6)	3289	3289.0	3289	3289.0	3294	3323.0	3289	3302.6	3289	3289.0	3289
n3m3P1S1r2-05	(53,6)	2578	2578.0	2578	2578.0	2580	2589.2	2578	2587.3	2578	2578.0	2581
n3m3P1S2r1-01	(53,6)	225	225.0	225	225.0	225	225.0	225	225.0	225	225.0	225
n3m3P1S2r1-02	(60,6)	583	583.0	583	583.0	583	583.0	583	583.0	583	583.0	583
n3m3P1S2r1-03	(74,6)	752	752.0	752	752.0	752	836.4	752	797.2	752	752.0	819
n3m3P1S2r1-04	(85,6)	989	989.0	989	989.0	1279	1422.4	989	990.2	989	989.0	989
n3m3P1S2r1-05	(64,6)	140	140.0	140	140.0	140	140.0	140	140.0	140	140.0	140
n3m3P1S2r2-01	(75,6)	3173	3173.0	3173	3173.0	3313	3491.8	3173	3286.0	3173	3173.0	3173
n3m3P1S2r2-02	(71,6)	3530	3530.0	3530	3530.0	3548	3660.0	3530	3546.2	3530	3530.0	3574
n3m3P1S2r2-03	(78,6)	2988	2988.0	2988	2988.0	2988	3406.0	2988	3633.5	2988	2988.0	4026
n3m3P1S2r2-04	(96,6)	5593	5593.0	5593	5596.0	6877	8978.2	5593	5681.5	5593	5593.0	6631
n3m3P1S2r2-05	(84,6)	6852	6852.0	6852	6853.3	9325	10532.8	6855	6940.4	6852	6852.0	7290
n3m3P2S1r1-01	(52,6)	1160	1160.0	1160	1160.0	1160	1182.5	1160	1211.0	1160	1160.0	1170
n3m3P2S1r1-02	(59,6)	1385	1385.0	1385	1385.0	1385	1385.0	1385	1385.0	1385	1385.0	1385
n3m3P2S1r1-03	(69,6)	1878	1878.0	1878	1878.0	1878	2076.5	1878	1893.0	1878	1878.0	1878
n3m3P2S1r1-04	(67,6)	1579	1579.0	1579	1579.0	1584	1604.8	1579	1634.8	1579	1579.0	1581
n3m3P2S1r1-05	(66,6)	1216	1216.0	1216	1217.3	1216	1218.4	1320	1400.0	1220	1219.8	1218
n3m3P2S1r2-01	(70,6)	6510	6510.0	6510	6510.0	6521	6525.8	6510	6510.0	6510	6510.0	6510
n3m3P2S1r2-02	(68,6)	4147	4147.0	4147	4147.0	4147	4187.0	4147	4192.0	4148	4148.3	4187
n3m3P2S1r2-03	(54,6)	6905	6905.0	6905	6905.0	6905	6920.0	6905	6963.8	6905	6905.0	7079
n3m3P2S1r2-04	(69,6)	8236	8236.0	8236	8236.0	8236	8284.0	8236	8289.3	8236	8236.0	8276
n3m3P2S1r2-05	(61,6)	6176	6176.0	6176	6176.0	6474	7144.5	6202	6476.7	6176	6176.0	6204
n3m3P2S2r1-01	(88,6)	1809	1809.0	1809	1810.0	2013	2112.2	1809	1824.4	1809	1809.0	1833
n3m3P2S2r1-02	(93,6)	2843	2843.0	2843	2843.0	2894	3069.6	2843	2843.0	2843	2843.0	2843
n3m3P2S2r1-03	(56,6)	1134	1134.0	1134	1134.0	1134	1134.0	1134	1134.0	1134	1134.0	1134
n3m3P2S2r1-04	(97,6)	3358	3358.0	3358	3358.0	3358	3559.8	3358	3358.0	3358	3358.0	3358
n3m3P2S2r1-05	(82,6)	1995	1995.0	1995	1995.0	1995	2017.2	1995	1995.0	1995	1995.0	1995
n3m3P2S2r2-01	(57,6)	6269	6269.0	6269	6270.3	6276	6352.0	6269	6276.0	6269	6269.0	6269
n3m3P2S2r2-02	(79,6)	9830	9830.0	9830	9830.0	10018	11248.0	9830	9830.0	9830	9830.0	9830
n3m3P2S2r2-03	(92,6)	16193	16193.0	16193	16193.0	18365	21614.8	16193	16482.4	16198	16197.8	17365
n3m3P2S2r2-04	(67,6)	5313	5313.0	5313	5313.0	5616	6556.4	5594	7088.0	5313	5313.0	5546
n3m3P2S2r2-05	(74,6)	9089	9089.0	9089	9089.0	9453	10782.0	9089	9572.8	9089	9089.0	9345
Average												

							, m=6							
ABC-TS			{RPD\vphantom{RPD}_{{Best\vphantom{Best}}}}						{RPD\vphantom{RPD}_{{Average\vphantom{Average}}}}					
Average	ET(sec)	Best of All	H + SA	H + VNS	PSO-GA	BKRGA-DE	IG	ABC-TS	H + SA	H + VNS	PSO-GA	BKRGA-DE	IG	ABC-TS
603.0	124.2	603	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.77	2.55	0.00	0.00
339.0	106.2	339	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
165.0	91.8	165	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	2.91	0.00	0.00	0.00
677.0	126	677	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.15	0.00	0.00	0.00
648.5	104.4	627	0.00	0.00	1.75	0.00	0.00	2.87	0.00	0.00	8.45	2.97	0.00	3.43
1948.0	104.4	1913	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.14	0.00	1.83
1935.0	106.2	1931	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.18	7.58	0.00	0.21
4507.3	151.2	4075	0.00	0.00	9.52	9.52	0.00	10.04	0.00	0.00	10.45	10.35	0.00	10.61
3297.5	99	3289	0.00	0.00	0.15	0.00	0.00	0.00	0.00	0.00	1.03	0.41	0.00	0.26
2594.3	95.4	2578	0.00	0.00	0.08	0.00	0.00	0.12	0.00	0.00	0.43	0.36	0.00	0.63
225.0	95.4	225	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
583.0	108	583	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
975.7	133.2	752	0.00	0.00	0.00	0.00	0.00	8.91	0.00	0.00	11.22	6.01	0.00	29.74
995.8	153	989	0.00	0.00	29.32	0.00	0.00	0.00	0.00	0.00	43.82	0.12	0.00	0.68
140.0	115.2	140	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3259.5	135	3173	0.00	0.00	4.41	0.00	0.00	0.00	0.00	0.00	10.05	3.56	0.00	2.73
3589.0	127.8	3530	0.00	0.00	0.51	0.00	0.00	1.25	0.00	0.00	3.68	0.46	0.00	1.67
4925.8	140.4	2988	0.00	0.00	0.00	0.00	0.00	34.74	0.00	0.00	13.99	21.60	0.00	64.85
7571.3	172.8	5593	0.00	0.00	22.96	0.00	0.00	18.56	0.00	0.05	60.53	1.58	0.00	35.37
8616.3	151.2	6852	0.00	0.00	36.09	0.04	0.00	6.39	0.00	0.02	53.72	1.29	0.00	25.75
1216.0	93.6	1160	0.00	0.00	0.00	0.00	0.00	0.86	0.00	0.00	1.94	4.40	0.00	4.83
1385.0	106.2	1385	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1878.0	124.2	1878	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.57	0.80	0.00	0.00
1585.7	120.6	1579	0.00	0.00	0.32	0.00	0.00	0.13	0.00	0.00	1.63	3.53	0.00	0.42
1219.0	118.8	1216	0.00	0.00	0.00	8.55	0.31	0.16	0.00	0.11	0.20	15.13	0.31	0.25
6510.0	126	6510	0.00	0.00	0.17	0.00	0.00	0.00	0.00	0.00	0.24	0.00	0.00	0.00
4193.7	122.4	4147	0.00	0.00	0.00	0.00	0.03	0.96	0.00	0.00	0.96	1.09	0.03	1.13
7079.0	97.2	6905	0.00	0.00	0.00	0.00	0.00	2.52	0.00	0.00	0.22	0.85	0.00	2.52
8289.3	124.2	8236	0.00	0.00	0.00	0.00	0.00	0.49	0.00	0.00	0.58	0.65	0.00	0.65
6280.0	109.8	6176	0.00	0.00	4.83	0.42	0.00	0.45	0.00	0.00	15.68	4.87	0.00	1.68
1852.3	158.4	1809	0.00	0.00	11.28	0.00	0.00	1.33	0.00	0.06	16.76	0.85	0.00	2.40
2843.0	167.4	2843	0.00	0.00	1.79	0.00	0.00	0.00	0.00	0.00	7.97	0.00	0.00	0.00
1134.0	100.8	1134	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3358.0	174.6	3358	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.01	0.00	0.00	0.00
1995.0	147.6	1995	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.11	0.00	0.00	0.00
6272.0	102.6	6269	0.00	0.00	0.11	0.00	0.00	0.00	0.00	0.02	1.32	0.11	0.00	0.05
9838.0	142.2	9830	0.00	0.00	1.91	0.00	0.00	0.00	0.00	0.00	14.43	0.00	0.00	0.08
17564.0	165.6	16193	0.00	0.00	13.41	0.00	0.03	7.24	0.00	0.00	33.48	1.79	0.03	8.47
6348.0	120.6	5313	0.00	0.00	5.70	5.29	0.00	4.39	0.00	0.00	23.40	33.41	0.00	19.48
11037.7	133.2	9089	0.00	0.00	4.00	0.00	0.00	2.82	0.00	0.00	18.63	5.32	0.00	21.44
			0.00	0.00	3.71	0.6	0.01	2.61	0.00	0.01	9.6	3.29	0.01	6.03



(a) The average RPD_{Best}



(b) The average $RPD_{Average}$

Fig. 8. The average RPDs of the proposed solution methods versus different problem categories.

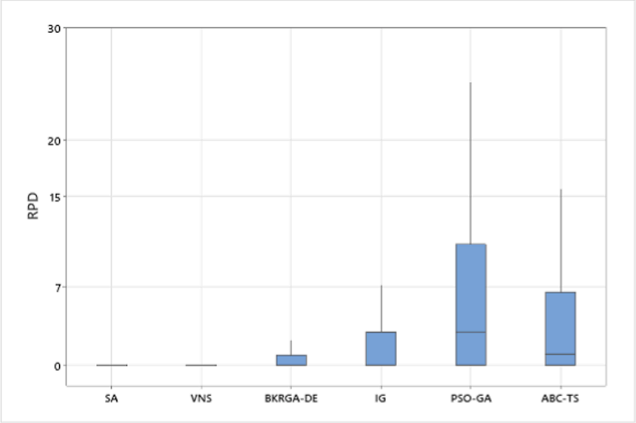


Fig. 9. Comparison of meta-heuristics: Boxplot.

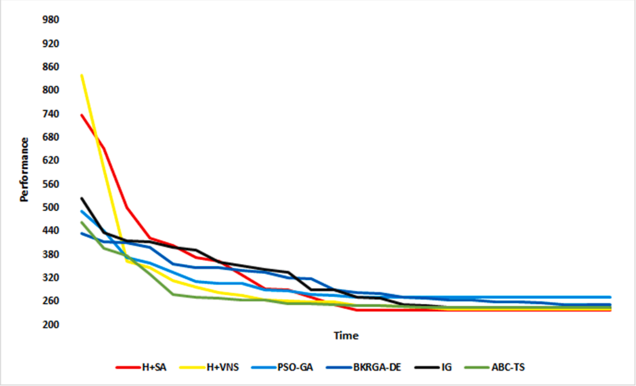


Fig. 10. Convergence status for n = 27 and m = 2.

Table 11

Kruskal–Wallis Test on RPD_{Best} values of different algorithms for small-sized instances.

Algorithm	N	Medians	Ave Rank	Z
ABC-TS	40	1.11125	147.8	2.73
BKRGA-DE	40	0.34584	116.5	−0.40
IG	40	0.63291	130.6	1.00
PSO-GA	40	0.34584	125.3	0.48
H + SA	40	0.00000	101.1	−1.94
H + VNS	40	0.00000	101.8	−1.87
Overall	240		120.5	
DF = 5	H-Value = 13.39	P-Value = 0.020		
DF = 5	H-Value = 14.54	P-Value = 0.013 (Adjusted for ties)		

Table 12

Kruskal–Wallis Test on RPD_{Best} values of different algorithms for medium-sized instances.

Algorithm	N	Medians	Ave Rank	Z
ABC-TS	80	0.50288	296.4	3.95
BKRGA-DE	80	0.00000	219.3	−1.50
IG	80	0.00000	245.8	0.38
PSO-GA	80	1.82609	331.6	6.44
H + SA	80	0.00000	160.1	−5.68
H + VNS	80	0.00000	189.8	−3.58
Overall	480		240.5	
DF = 5	H-Value = 87.09	P-Value = 0.000		
DF = 5	H-Value = 114.12	P-Value = 0.000 (Adjusted for ties)		

Table 13

Kruskal–Wallis Test on RPD_{Best} values of different algorithms for large-sized instances.

Algorithm	N	Medians	Ave Rank	Z
ABC-TS	120	1.35489	453.2	5.35
BKRGA-DE	120	0.00000	351.7	−0.51
IG	120	0.00000	366.4	0.34
PSO-GA	120	5.42699	513.6	8.84
H + SA	120	0.00000	215.5	−8.37
H + VNS	120	0.00000	262.5	−5.66
Overall	720		360.5	
DF = 5	H-Value = 174.21	P-Value = 0.000		
DF = 5	H-Value = 221.36	P-Value = 0.000 (Adjusted for ties)		

Table 14
Comparison between SA and VNS: Test of Mann Whitney.

Algorithm	*N*	*Medians*
H + SA	240	0.0000
H + VNS	240	0.0000

Point estimate for $\eta_1 - \eta_2$ is −0.00000
95.0 Percent CI for $\eta_1 - \eta_2$ is (-0.00000, −0.00000)
W = 53767.50
Test of $\eta_1 = \eta_2$ vs $\eta_1 \neq \eta_2$ is significant at 0.009
The test is significant at 0.0000 (adjusted for ties)

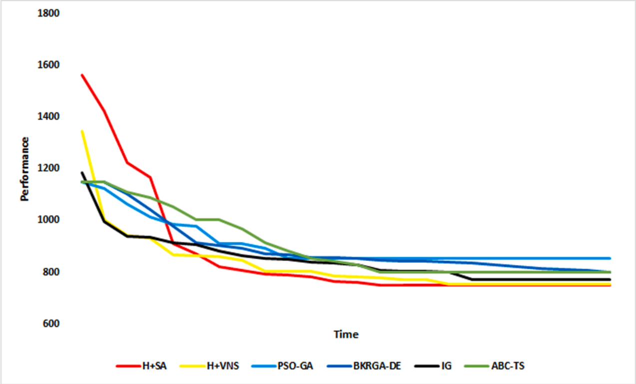


Fig. 11. Convergence status for n = 40 and m = 4.

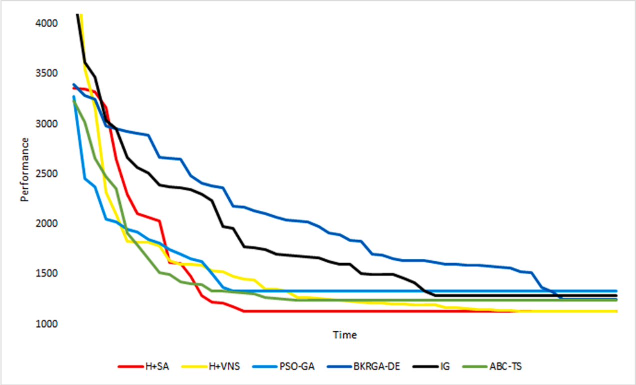


Fig. 12. Convergence status for n = 72 and m = 2.

7. Conclusion

Joint scheduling jobs and maintenance activity is a challenging work in manufacturing systems. This research domain aims to develop constructive methods for scheduling jobs when a production halt because of maintenance happens. In this research, we have studied a heterogeneous parallel machines BP problem with release dates, due dates, and variable maintenance operation to minimize total tardiness $((R_m|p-batch, MA, r_j|\sum_{j=1}^{n} t_j))$. In order to find optimal solutions for the small-scale test problems, an MILP formulation has been proposed. Since mathematical formulation is not capable of dealing with the test instances with medium and large scale due to the NP-hard nature of the problem, at first, a heuristic approach has been developed to find feasible solutions. Then, two meta-heuristics based on H + SA and H + VNS have been proposed to solve the problem with different sizes. To evaluate the performance of the proposed solution methods, four meta-heuristics, including PSO-GA by Beldar and Costa (2018), BKRGA-DE by Kong et al. (2020), ABC-TS by Lu et al. (2018), and IG by Arroyo et al. (2019), have been adopted from the literature of relevant research studies and their results were compared with our solution approaches. Firstly, the Kruskal–Wallis test on RPDs was used to compare the medians of the six solution methods so as to find the best performing approach. The results of Kruskal–Wallis test for different categories of test instances show that H + SA and H + VNS are obviously superior to other four solution methods. Hence, in order to discover the approach with the best performance between H + SA and H + VNS, Mann–Whitney test has been applied. The test results demonstrate that H + SA had a better performance than H + VNS. As the total increase in tardiness of jobs can lead to the loss of goodwill for the organization as well as the compensation payment to customers, the outcomes of this research can help the manufacturing companies in this regard. In order to consider more practical case in joint job scheduling with maintenance task, as a future study, it is useful to take into account the number of maintenance tasks performed on each machine as a decision variable and the order in which they should be performed. It would be also interesting to study the variable maintenance activity in other environments, including flow shop and job shop. In addition, a lower bounding technique could be developed to determine the effectiveness and efficiency of the proposed solution approaches.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Al-Salamah, M. (2015). Constrained binary artificial bee colony to minimize the makespan for single machine batch processing with non-identical job sizes. *Applied Soft Computing, 29*, 379–385.

Alfares, H., Mohammed, A., & Ghaleb, M. (2021). Two-machine scheduling with aging effects and variable maintenance activities. *Computers & Industrial Engineering, 160*, 107586.

Arroyo, J. E. C., & Leung, J. Y. T. (2017). An effective iterated greedy algorithm for scheduling unrelated parallel batch machines with non-identical capacities and unequal ready times. *Computers & Industrial Engineering, 105*, 84–100.

Arroyo, J. E. C., Leung, J. Y. T., & Tavares, R. G. (2019). An iterated greedy algorithm for total flow time minimization in unrelated parallel batch machines with unequal job release times. *Engineering Applications of Artificial Intelligence, 77*, 239–254.

Avalos-Rosales, O., Angel-Bello, F., Álvarez, A., & Cardona-Valdés, Y. (2018). Including preventive maintenance activities in an unrelated parallel machine environment with dependent setup times. *Computers & Industrial Engineering, 123*, 364–377.

Baker, K., & Trietsch, D. (2009). *Principles of sequencing and scheduling*. John Wiley & Sons.

Beldar, P., & Costa, A. (2018). Single machine batch processing problem with release dates to minimize total completion time. *International Journal of Industrial Engineering Computations, 9*, 331–348.

Beldar, P., Framinan, J. M., & Ardakani, A. (2019). Minimization of total completion time on a batch processing machine with arbitrary release dates: an effectual teaching–learning based optimization approach. *Production Engineering, 13*, 557–566.

Bilyk, A., Mönch, L., & Almeder, C. (2014). Scheduling jobs with ready times and precedence constraints on parallel batch machines using metaheuristics. *Computers & Industrial Engineering, 78*, 175–185.

Cerny, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications, 45*, 41–51.

Chen, J. S. (2006a). Optimization models for the machine scheduling problem with a single flexible maintenance activity. *Engineering Optimization, 38*, 53–71.

Chen, J. S. (2006b). Single-machine scheduling with flexible and periodic maintenance. *Journal of the Operational Research Society, 57*, 703–710.

Chen, J. S. (2008). Scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan. *European Journal of Operational Research, 190*, 90–102.

Cheng, T., Hsu, C. J., & Yang, D. L. (2011). Unrelated parallel-machine scheduling with deteriorating maintenance activities. *Computers & Industrial Engineering, 60*, 602–605.

Cui, W. W., & Lu, Z. (2017). Minimizing the makespan on a single machine with flexible maintenances and jobs' release dates. *Computers & Operations Research, 80*, 11–22.

Damodaran, P., & Vélez-Gallego, M. C. (2012). A simulated annealing algorithm to minimize makespan of parallel batch processing machines with unequal job ready times. *Expert Systems with Applications, 39*, 1451–1458.

Detti, P., Nicosia, G., Pacifici, A., Manrique, Zabalo, & de Lara, G. (2019). Robust single machine scheduling with a flexible maintenance activity. *Computers & Operations Research, 107*, 19–31.

Ding, J., Schulz, S., Shen, L., Buscher, U., & Lü, Z. (2021). Energy aware scheduling in flexible flow shops with hybrid particle swarm optimization. *Computers & Operations Research, 125*, 105088.

Hansen, P., & Mladenovic, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research, 130*, 449–467.

Hansen, P., Mladenovic, N., & Pérez, J. A. M. (2008). Variable neighbourhood search: methods and applications. *4OR, 6*, 319–360.

Hsu, C. J., Ji, M., Guo, J. Y., & Yang, D. L. (2013). Unrelated parallel-machine scheduling problems with aging effects and deteriorating maintenance activities. *Information Sciences, 253*, 163–169.

Huang, J., Wang, L., 2018. Makespan minimization on single batch-processing machine considering preventive maintenance, pp. 294–298.

Huang, J., Wang, L., & Jiang, Z. (2020). A method combining rules with genetic algorithm for minimizing makespan on a batch processing machine with preventive maintenance. *International Journal of Production Research, 58*, 4086–4102.

Hulett, M., Damodaran, P., & Amouie, M. (2017). Scheduling non-identical parallel batch processing machines to minimize total weighted tardiness using particle swarm optimization. *Computers & Industrial Engineering, 113*, 425–436.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*, 671–680.

Kong, M., Liu, X., Pei, J., Cheng, H., & Pardalos, P. (2020). A brkga-de algorithm for parallel-batching scheduling with deterioration and learning effects on parallel machines under preventive maintenance consideration. *Annals of Mathematics and Artificial Intelligence, 88*, 237–267.

Kumar Manjeshwar, P., Damodaran, P., & Srihari, K. (2009). Minimizing makespan in a flow shop with two batch-processing machines using simulated annealing. *Robotics and Computer-Integrated Manufacturing, 25*, 667–679.

Lee, C. Y. (1999). Minimizing makespan on a single batch processing machine with dynamic job arrivals. *International Journal of Production Research, 37*, 219–236.

Lei, D., & Guo, X. P. (2011). Variable neighbourhood search for minimising tardiness objectives on flow shop with batch processing machines. *International Journal of Production Research, 49*, 519–529.

Li, D., Meng, X., Liang, Q., & Zhao, J. (2015). A heuristic-search genetic algorithm for multi-stage hybrid flow shop scheduling with single processing machines and batch processing machines. *Journal of Intelligent Manufacturing, 26*, 873–890.

Lin, S. W., Cheng, C. Y., Pourhejazy, P., & Ying, K. C. (2021). Multi-temperature simulated annealing for optimizing mixed-blocking permutation flowshop scheduling problems. *Expert Systems with Applications, 165*, 113837.

Liu, M., Wang, S., Chu, C., & Chu, F. (2016). An improved exact algorithm for single-machine scheduling to minimise the number of tardy jobs with periodic maintenance. *International Journal of Production Research, 54*, 3591–3602.

Low, C., Ji, M., Hsu, C. J., & Su, C. T. (2010). Minimizing the makespan in a single machine scheduling problems with flexible and periodic maintenance. *Applied Mathematical Modelling, 34*, 334–342.

Lu, S., Liu, X., Pei, J., T. Thai, M., M. Pardalos, P., 2018. A hybrid abc-ts algorithm for the unrelated parallel-batching machines scheduling problem with deteriorating jobs and maintenance activity. Applied Soft Computing 66, 168–182.

Luo, W., Cheng, T., & Ji, M. (2015). Single-machine scheduling with a variable maintenance activity. *Computers & Industrial Engineering, 79*, 168–174.

Luo, W., & Ji, M. (2015). Scheduling a variable maintenance and linear deteriorating jobs on a single machine. *Information Processing Letters, 115*, 33–39.

Mann, H. B., & Whitney, D. (1947). On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics, 18*, 50–60.

Mladenovic, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research, 24*, 1097–1100.

Pacheco, J., Porras, S., Casado, S., & Baruque, B. (2018). Variable neighborhood search with memory for a single-machine scheduling problem with periodic maintenance and sequence-dependent set-up times. *Knowledge-Based Systems, 145*, 236–249.

Pinedo, M. L. (2012). *Scheduling: Theory, Algorithms, and Systems*. New York: Springer-Verlag.

Ruiz-Torres, A. J., Paletta, G., & M'Hallah, R. (2017). Makespan minimisation with sequence-dependent machine deterioration and maintenance events. *International Journal of Production Research, 55*, 462–479.

Sbihi, M., & Varnier, C. (2008). Single-machine scheduling with periodic and flexible periodic maintenance to minimize maximum tardiness. *Computers & Industrial Engineering, 55*, 830–840.

Shahvari, O., & Logendran, R. (2017). An enhanced tabu search algorithm to minimize a bi-criteria objective in batching and scheduling problems on unrelated-parallel machines with desired lower bounds on batch sizes. *Computers & Operations Research, 77*, 154–176.

Sun, X., & Geng, X. N. (2019). Single-machine scheduling with deteriorating effects and machine maintenance. *International Journal of Production Research, 57*, 3186–3199.

Tan, Y., Mönch, L., & Fowler, J. W. (2018). A hybrid scheduling approach for a two-stage flexible flow shop with batch processing machines. *Journal of Scheduling, 21*, 209–226.

Uzsoy, R. (1994). Scheduling a single batch processing machine with non-identical job sizes. *International Journal of Production Research, 32*, 1615–1635.

Wang, H. M., & Chou, F. D. (2010). Solving the parallel batch-processing machines with different release times, job sizes, and capacity limits by metaheuristics. *Expert Systems with Applications, 37*, 1510–1521.

Wang, J. B., & Wei, C. M. (2011). Parallel machine scheduling with a deteriorating maintenance activity and total absolute differences penalties. *Applied Mathematics and Computation, 217*, 8093–8099.

Wang, Q., Liu, A., & Xiao, J. (2017). On exact algorithms for single-machine scheduling problems with a variable maintenance. *Computers & Industrial Engineering, 107*, 276–279.

Wang, T., Baldacci, R., Lim, A., & Hu, Q. (2018). A branch-and-price algorithm for scheduling of deteriorating jobs and flexible periodic maintenance on a single machine. *European Journal of Operational Research, 271*, 826–838.

Wang, Y., Li, T., Wang, B., Wu, Z., 2019. Hot-Rolled Batch Scheduling of Seamless Steel Tube with Flexible Machine Maintenance: Proceedings of ICCD 2017. pp. 773–778.

Wu, C. C., Bai, D., Chen, J. H., Lin, W. C., Xing, L., Lin, J. C., & Cheng, S. R. (2021). Several variants of simulated annealing hyper-heuristic for a single-machine scheduling with two-scenario-based dependent processing times. *Swarm and Evolutionary Computation, 60*, 100765.

Xu, D., & Yin, Y. (2011). On single-machine scheduling with flexible maintenance activities. *The International Journal of Advanced Manufacturing Technology, 56*, 1139–1145.

Xu, D., Yin, Y., & Li, H. (2010). Scheduling jobs under increasing linear machine maintenance time. *J. Scheduling, 13*, 443–449.

Xu, S., Dong, W., Jin, M., & Wang, L. (2020). Single-machine scheduling with fixed or flexible maintenance. *Computers & Industrial Engineering, 139*, 106203.

Yang, D. L., Hung, C. L., Hsu, C. J., & Chern, M. S. (2002). Minimizing the makespan in a single machine scheduling problem with a flexible maintenance. *Journal of the Chinese Institute of Industrial Engineers, 19*, 63–66.

lin Yang, S., Ma, Y., ling Xu, D., bo Yang, J., 2011. Minimizing total completion time on a single machine with a flexible maintenance activity. Computers & Operations Research 38, 755–770.

Yang, S. J., & Yang, D. L. (2010). Minimizing the makespan on single-machine scheduling with aging effect and variable maintenance activities. *Omega, 38*, 528–533.

Ying, K. C., & Lin, S. W. (2020). Solving no-wait job-shop scheduling problems using a multi-start simulated annealing with bi-directional shift timetabling algorithm. *Computers & Industrial Engineering, 146*, 106615.

Ying, K. C., Lu, C. C., & Chen, J. C. (2016). Exact algorithms for single-machine scheduling problems with a variable maintenance. *Computers & Industrial Engineering, 98*, 427–433.

Yoo, J., & Lee, I. S. (2016). Parallel machine scheduling with maintenance activities. *Computers & Industrial Engineering, 101*, 361–371.

Zammori, F., Braglia, M., & Castellano, D. (2014). Harmony search algorithm for single-machine scheduling problem with planned maintenance. *Computers & Industrial Engineering, 76*, 333–346.

Zarook, Y., Rezaeian, J., Tavakkoli-Moghaddam, R., Mahdavi, I., & Javadian, N. (2014). Minimization of makespan for the single batch-processing machine scheduling problem with considering aging effect and multi-maintenance activities. *The International Journal of Advanced Manufacturing Technology, 76*, 1879–1892.

Zhang, X., Liu, S. C., Lin, W. C., & Wu, C. C. (2020). Parallel-machine scheduling with linear deteriorating jobs and preventive maintenance activities under a potential machine disruption. *Computers & Industrial Engineering, 145*, 106482.

Zhou, H., Pang, J., Chen, P. K., & Chou, F. D. (2018a). A modified particle swarm optimization algorithm for a batch-processing machine scheduling problem with arbitrary release times and non-identical job sizes. *Computers & Industrial Engineering, 123*, 67–81.

Zhou, S., Liu, M., Chen, H., & Li, X. (2016). An effective discrete differential evolution algorithm for scheduling uniform parallel batch processing machines with non-identical capacities and arbitrary job sizes. *International Journal of Production Economics, 179*, 1–11.

Zhou, S., Xie, J., Du, N., & Pang, Y. (2018b). A random-keys genetic algorithm for scheduling unrelated parallel batch processing machines with different capacities and arbitrary job sizes. *Applied Mathematics and Computation, 334*, 254–268.

Zhou, S., Xing, L., Zheng, X., Du, N., Wang, L., & Zhang, Q. (2021). A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times. *IEEE Transactions on Cybernetics, 51*, 1430–1442.