

Research Article

Hybrid Metaheuristics for Solving a Fuzzy Single Batch-Processing Machine Scheduling Problem

S. Molla-Alizadeh-Zavardehi,¹ R. Tavakkoli-Moghaddam,² and F. Hosseinzadeh Lotfi³

¹ Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

² School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

³ Department of Mathematics, Science and Research Branch, Islamic Azad University, Tehran, Iran

Correspondence should be addressed to S. Molla-Alizadeh-Zavardehi; saber.alizadeh@gmail.com

Received 14 February 2014; Accepted 30 March 2014; Published 22 April 2014

Academic Editors: P. Agarwal, V. Bhatnagar, and Y. Zhang

Copyright © 2014 S. Molla-Alizadeh-Zavardehi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper deals with a problem of minimizing total weighted tardiness of jobs in a real-world single batch-processing machine (SBPM) scheduling in the presence of fuzzy due date. In this paper, first a fuzzy mixed integer linear programming model is developed. Then, due to the complexity of the problem, which is NP-hard, we design two hybrid metaheuristics called GA-VNS and VNS-SA applying the advantages of genetic algorithm (GA), variable neighborhood search (VNS), and simulated annealing (SA) frameworks. Besides, we propose three fuzzy earliest due date heuristics to solve the given problem. Through computational experiments with several random test problems, a robust calibration is applied on the parameters. Finally, computational results on different-scale test problems are presented to compare the proposed algorithms.

1. Introduction

A batch-processing machine (BPM) is a special variant of a scheduling problem, in which several jobs can be simultaneously processed in such a way that all the jobs in a batch start and complete their processing at the same time. The main advantage is to reduce setups and/or facilitation of material handling. The problem of BPM scheduling is often encountered in real industries. The industrial application of these machines can be found in semiconductor burn-in operations, environmental stress-screening (ESS) chambers, chemical, food, and mineral processing, pharmaceutical and construction materials industries, and so forth.

The BPM scheduling problem is important because the scheduling of batching operations has a significant economic impact. It is mainly motivated by an industrial application, namely, the burn-in operation found in the final testing phase in semiconductor manufacturing [1, 2]. In the semiconductor manufacturing, the jobs have different processing times and

sizes that are both required by the customers. The jobs are grouped in batches where a batch means a subset of jobs. The BPM can process a batch of jobs as long as the sum of all the job sizes in the batch does not violate the capacity of the machine. The processing time of a batch is equal to the longest processing time of all the jobs in that batch.

Ikura and Gimple [3] were the first researchers who studied the BPM problem and Lee et al. [4] first presented a detailed description for burn-in operation. As reported in the studies, the exact algorithms have a slow convergence rate and they can solve only small instances to optimality.

As this study addresses SBPM with fuzzy due dates using metaheuristics, the review on SBPM scheduling under a fuzzy environment and the application of metaheuristics to these problems is carried out. For an extensive review on BPM scheduling problems, we refer to Potts and Kovalyov [5] and Mathirajan and Sivakumar [6].

In BPM scheduling problems, Wang and Uzsoy [7] firstly proposed a metaheuristic algorithm. Considering dynamic

job arrivals, they combined a dynamic programming algorithm with a random key genetic algorithm (GA) to minimize the maximum lateness. Melouk et al. [8] used a simulated annealing (SA) to minimize the makespan. Koh et al. [9] proposed a random key representation-based GA for the problems of minimizing the makespan and total weighted completion time. Sevaux and Dauzère-Pérès [10], Hussein-zadeh Kashan et al. [11], and Damodaran et al. [12] used a GA and redesigned the coding and decoding methods.

Mönch et al. [13] presented a GA combined with dominance properties to minimize the earliness tardiness of the jobs. Chou et al. [14] and Wang et al. [15] presented a hybrid GA and a hybrid forward/backward approach to minimize the makespan. Kashan and Karimi [16] developed two versions of an ant colony optimization (ACO) framework under the situation considered in Koh et al. [9]. Chou and Wang [17], Mathirajan et al. [18], and Wang [19] proposed a hybrid GA, SA, and iterated heuristic for the objective of the total weighted tardiness, respectively. Hussein-zadeh Kashan et al. [20] considered bicriteria scheduling for the simultaneous minimization of the makespan and maximum tardiness.

In the classic scheduling problems, it is usually assumed that the aspects of the problem in hand are certain. Most existing models neglect the presence of uncertainty within a scheduling environment. In many real-world scheduling problems, however, uncertainty and vagueness in due date often do exist that make the models more complex. This uncertainty may come about because of production problems (e.g., defect in raw material and machine malfunctioning) or problems with delivery itself (e.g., transportation delay and traffic jam). Although classic BPM scheduling models are extensively studied in the literature, there are only three studies on fuzzy-based BPM models.

Ishii et al. [21] introduced the concept of fuzzy due dates to scheduling problems; fuzzy due dates scheduling problems have been investigated by many researchers. Harikrishnan and Ishii [22] presented a polynomial time algorithm for bicriteria scheduling of serial-batching problem with fuzzy due dates to minimize the total weighted resource consumption and maximize the minimal satisfaction degree. Yimer and Demirli [23] considered a fuzzy goal programming problem for batch scheduling of jobs in a two-stage flow shop to minimize the total weighted flow time of jobs. Cheng et al. [24] proposed ACO to minimize the fuzzy makespan on an SBPM with triangular fuzzy processing times.

Till now, none has considered the objective of minimizing the fuzzy total weighted fuzzy tardiness penalties. So, a new approach to solve a fuzzy SBPM (FSBPM) is proposed and a related fuzzy number is considered for due dates and modeled by fuzzy sets, in which the corresponding membership functions represent satisfaction degree with respect to jobs' completion times. Hence, for the first time, we present a new programming approach. Since the problem is NP-hard for solving the addressed problem, two hybrid metaheuristics (GA-VNS and VNS-SA) are developed to obtain better results.

The remainder of this paper is as follows. Section 2 describes the problem in detail and presents the fuzzy mathematical model. Section 3 explains the proposed algorithms. Section 4 describes the experimental design and compares the computational results. Finally, conclusions are provided and some areas of further research are then suggested in Section 5.

2. Fuzzy Mathematical Model and Problem Descriptions

2.1. Deterministic Model. The objective of this problem is to minimize the total weighted tardiness penalties. Suppose that there are n jobs to be processed and each job $j \in J$ has a processing time p_j and a corresponding size s_j . The total size of all the jobs in a batch does not exceed machine capacity S . The processing time of a batch b is given by the longest job in the batch (i.e., $P^b = \max\{p_j \mid j \in \text{batch } b\}$). The formulation is as follows.

Notations

Sets

J : Jobs, $j \in J$

B : Batches, $b \in B$.

Parameters

p_j : Processing time of job j

s_j : Size of job j

cap: Machine capacity

β_j : Tardiness penalty (/unit/h) of job j

d_j : Due date of job j .

Decision Variables

X_{jb} : A binary variable indicates the assignment of job j to batch b

P^b : Processing time of batch b

c_j : Completion time of job j

C^b : Completion time of batch b

T_j : Tardiness of job j .

According to the mentioned sets, parameters, and decision variables, the mathematical formulation of the total weighted tardiness penalties can be written below:

$$\text{Min } Z = \sum_{j \in J} \beta_j T_j s_j \quad (1)$$

$$\text{s.t: } \sum_{b \in B} X_{jb} = 1 \quad \forall j \in J \quad (2)$$

$$\sum_{j \in J} s_j X_{jb} \leq \text{cap} \quad \forall b \in B \quad (3)$$

$$P^b \geq p_j X_{jb} \quad \forall j \in J, \forall b \in B \tag{4}$$

$$C^b = \sum_{i=1}^b P^i \quad \forall b \in B \tag{5}$$

$$c_j \geq C^b - M(1 - X_{jb}) \quad \forall j \in J, \forall b \in B; \tag{6}$$

M is a very large positive number

$$T_j \geq c_j - d_j \quad \forall j \in J \tag{7}$$

$$X_{jb} \in \{0, 1\} \quad \forall j \in J, \forall b \in B. \tag{8}$$

The objective function is to minimize the total weighted tardiness penalties of jobs. Constraint set (2) ensures that each job can be processed in only one batch. Constraint set (3) ensures that the machine capacity is not exceeded when jobs are assigned to a batch. Constraint set (4) states that the processing time of a batch is the longest processing time among all the jobs in that batch. Constraint set (5) determines the completion time of each batch. Constraint set (6) defines the completion time of each job as the completion time of the batch that it is processed in. Constraint set (7) defines the tardiness of a job as the difference between the due date of a job and its completion time or 0 if it is negative. Constraint set (8) specifies the type of decision variable X_{jb} .

Due to minimization of just only tardiness or total weighted tardiness penalties in the objective function, the model chooses the minimum P^b in the constraint sets (4) to reach the longest processing time among all the jobs in that batch. The smaller the completion time of jobs, the more desirable the objective function. Similarly, the model finds the minimum c_j and T_j in the constraint sets (6) and (7).

2.2. Fuzzy Model. We briefly introduce some basic concepts and results about fuzzy measure theory.

Definition 1. If X is a collection of objects denoted generically by x , then a fuzzy set in X is a set of the ordered pairs:

$$\tilde{d} = \{x, \tilde{d}(x) \mid x \in X\}, \tag{9}$$

where $\tilde{d}(x)$ is called the membership function that is associated with each $x \in X$ a number in $[0, 1]$ indicating to what degree x is a number.

Definition 2. $\tilde{d}_j = \{d_{j,l}, d_{j,u}\}$ denotes a fuzzy number as shown in Figure 1.

As mentioned in the literature, the concept of fuzzy due dates has been used in scheduling problems. Here, this concept is being firstly utilized in the BPM scheduling problem. In a fuzzy due date, the membership function assigned to each job represents the customer satisfaction degree for

the delivery or completion time of that job. The membership function of a fuzzy due date of a job is represented below:

$$\mu_j(C_j) = \begin{cases} 1 & \text{if } c_j \leq d_{j,l} \\ \frac{d_{j,u} - c_j}{d_{j,u} - d_{j,l}} & \text{if } d_{j,l} < c_j < d_{j,u} \\ 0 & \text{if } c_j \geq d_{j,u}. \end{cases} \tag{10}$$

From Figure 1, we can see that the full satisfaction (i.e., $\mu_j(C_j) = 1$) is attained if $c_j \leq d_{j,l}$, and the satisfaction grade is positive if $d_{j,l} < c_j < d_{j,u}$ in the membership function (8). If $d_{j,l} = d_{j,u}$, the fuzzy due date is transformed to interval due date or due window.

According to the mentioned fuzzy due date, the studied problem can be formulated as a maximization problem of the total degree of satisfaction over given jobs or equivalently a minimization problem of the total degree of dissatisfaction. For the fuzzy mathematical formulation, the objective function (11) and constraint sets (12) and (13) are replaced instead of objective function (1) and constraint set (7) to calculate the total degree of satisfaction:

$$\text{Max } Z = \sum_{j \in J} w_j \mu_j s_j \tag{11}$$

$$\mu_j = 1 \quad \text{if } c_j \leq d_{j,l} \tag{12}$$

$$\mu_j = \frac{d_{j,u} - c_j}{d_{j,u} - d_{j,l}} \quad \text{if } d_{j,l} < c_j < d_{j,u}. \tag{13}$$

We can also use the following objective function (20) to calculate the total degree of satisfaction instead of expressions (11)–(13):

$$\text{Max } Z = \sum_{j \in J} w_j s_j \left(\left(\frac{\max(0, d_{j,l} - c_j)}{d_{j,l} - c_j} \right) + \left(\frac{\max(0, c_j - d_{j,l})}{c_j - d_{j,l}} \right) \times \left(\frac{\max(0, d_{j,u} - c_j)}{d_{j,u} - c_j} \right) \times \left(\frac{d_{j,u} - c_j}{d_{j,u} - d_{j,l}} \right) \right). \tag{14}$$

It is clear that $\max(0, d_{j,l} - c_j)/(d_{j,l} - c_j) = 1$ results in $c_j \leq d_{j,l}$, while $\max(0, c_j - d_{j,l})/(c_j - d_{j,l})$ and $\max(0, d_{j,u} - c_j)/(d_{j,u} - c_j) = 1$ result in $d_{j,l} < c_j < d_{j,u}$. For more explanation of $(\max(0, d_{j,l} - c_j)/(d_{j,l} - c_j)) + (\max(0, c_j - d_{j,l})/(c_j - d_{j,l}))(\max(0, d_{j,u} - c_j)/(d_{j,u} - c_j))((d_{j,u} - c_j)/(d_{j,u} - d_{j,l}))$,

consider a simple example of $\tilde{d}_j = \{d_{j,l}, d_{j,u}\} = \{4, 7\}$ with different completion times:

$$\begin{aligned}
 (1) \quad c_j = 3 &\implies \left(\frac{\max(0, 4 - 3)}{4 - 3}\right) + \left(\frac{\max(0, 3 - 4)}{3 - 4}\right) \\
 &\quad \times \left(\frac{\max(0, 7 - 3)}{7 - 3}\right) \left(\frac{7 - 3}{7 - 4}\right) = 1 + 0 \times 1 \times \frac{4}{3} = 1. \\
 (2) \quad c_j = 5 &\implies \left(\frac{\max(0, 4 - 5)}{4 - 5}\right) + \left(\frac{\max(0, 5 - 4)}{5 - 4}\right) \\
 &\quad \times \left(\frac{\max(0, 7 - 5)}{7 - 5}\right) \left(\frac{7 - 5}{7 - 4}\right) = 0 + 1 \times 1 \times \frac{2}{3} = \frac{2}{3}. \\
 (3) \quad c_j = 8 &\implies \left(\frac{\max(0, 4 - 8)}{4 - 8}\right) + \left(\frac{\max(0, 8 - 4)}{8 - 4}\right) \\
 &\quad \times \left(\frac{\max(0, 7 - 8)}{7 - 8}\right) \left(\frac{7 - 8}{7 - 4}\right) = 0 + 1 \times 0 \times \frac{-1}{3} = 0. \tag{15}
 \end{aligned}$$

As mentioned above, similar to the expressions (11)–(14), expressions (1) and (16)–(18) can be used for the equivalent fuzzy mathematical formulation of the total degree of dissatisfaction as follows:

$$T_j = \frac{c_j - d_{j,l}}{d_{j,u} - d_{j,l}} \quad \text{if } d_{j,l} < c_j < d_{j,u} \tag{16}$$

$$T_j = 1 \quad \text{if } c_j \geq d_{j,u} \tag{17}$$

$$\begin{aligned}
 \text{Min } Z &= \sum_{j \in J} \beta_j s_j \left(\left(\frac{\max(0, c_j - d_{j,l})}{c_j - d_{j,l}} \right) \right. \\
 &\quad \times \left(\frac{\max(0, d_{j,u} - c_j)}{d_{j,u} - c_j} \right) \\
 &\quad \times \left(\frac{c_j - d_{j,l}}{d_{j,u} - d_{j,l}} \right) \\
 &\quad \left. + \left(\frac{\max(0, c_j - d_{j,u})}{c_j - d_{j,u}} \right) \right). \tag{18}
 \end{aligned}$$

Linearization. Obviously, the proposed fuzzy model is a nonlinear mathematical model because of the conditional expressions in the constraint sets (12), (13), (16), and (17). Also, multiplication of variables and max function in the objective functions (14) and (18) are used. An attempt is made in this part to linearize the fuzzy model via introducing

binary variable. Hence, the following constraints should be used instead of nonlinear constraint sets (12) and (13):

$$d_{j,u} - c_j \geq M(y_j - 1) \quad \forall j \in J \tag{19}$$

$$\mu_j \leq \frac{d_{j,u} - c_j}{d_{j,u} - d_{j,l}} + M(1 - y_j) \quad \forall j \in J \tag{20}$$

$$\mu_j \leq y_j \quad \forall j \in J \tag{21}$$

$$y_j \in \{0, 1\} \quad \forall j \in J. \tag{22}$$

Similarly, the constraint (22), following objective function and constraints, should be used instead of objective function (1) and nonlinear constraint sets (16) and (17):

$$\begin{aligned}
 \text{Min } Z &= \sum_{j \in J} \beta_j (T_{j,1} + T_{j,2}) s_j \\
 c_j - d_{j,u} &\leq M T_{j,1} \quad \forall j \in J \\
 d_{j,u} - c_j &\leq M y_j \quad \forall j \in J \tag{23}
 \end{aligned}$$

$$T_{j,2} \geq \frac{c_j - d_{j,l}}{d_{j,u} - d_{j,l}} - M(1 - y_j) \quad \forall j \in J.$$

3. Solution Approach

The evolutionary computation community has shown for many years significant interest in optimization problems, in particular in the global optimization of real valued problems, for which exact and analytical methods are not productive. These techniques have shown great promise in several real-world applications [25, 26]. Hence, these methods are often utilized in order to solve the problem in a shorter run time.

3.1. Proposed Earliest Due Date Heuristics. In this subsection, we propose three constructive greedy heuristics based on EDD as a well-known heuristic method related to the due date. The details of these proposed heuristics are as follows.

- (i) Calculate the index of jobs to be scheduled.
- (ii) Sort jobs in increasing order of their index.
- (iii) Apply the first-first (FF) heuristic to group jobs into batches.

Accordingly, the details of these three variants of EDDs are as follows:

EDD Algorithm. In this variant, the indexes are equal to the EDD of the respective jobs. The centroid-based distance method is used for ranking fuzzy numbers as follows:

$$\begin{aligned}
 \text{Crisp due date } (d) &= \frac{\int_0^{d_l} x \, dx + \int_{d_l}^{d_u} x \left(\frac{(d_u - x)}{(d_u - d_l)} \right) dx}{\int_0^{d_l} dx + \int_{d_l}^{d_u} \left(\frac{(d_u - x)}{(d_u - d_l)} \right) dx} \tag{24} \\
 &= \frac{1}{3} \left(d_l + d_u - \frac{d_l d_u}{d_l + d_u} \right).
 \end{aligned}$$

The jobs are sorted in increasing order of their crisp due dates. So, the job that has the earliest due date will be allotted first.

EDDL Algorithm. Sort jobs in increasing order of their d_i .

EDDU Algorithm. Sort jobs in increasing order of their d_u .

3.2. Encoding Scheme and Initialization. As mentioned earlier in the literature, the random key (RK) method is used for solving BPM scheduling problems. To generate a sequence by this method, random real numbers between zero and one are generated for each job. By ascending sorting of the value corresponding to each job, the sequence of job is obtained and then the FF heuristic is applied to group the jobs into the batches. After having a permutation and forming the batches, we can use it to compute the objective function value of this solution.

3.3. Hybrid Metaheuristics. Over the last years, considerable research has been conducted in hybrid metaheuristics in the field of optimization. The trade-off between intensification and diversification mechanisms is the main aspect of these algorithms. Generally, metaheuristics can be categorized into two main classes: local search methods and population based methods. Population based methods deal with a set of solutions in every iteration of the algorithm, while local search heuristics only deal with a single solution.

Although local search heuristics only deal with a single solution, it has shown its potential in both exploring and exploiting the promising regions in the search space with high quality solutions such as VNS. On the other hand, the basic scheme of VNS and its extensions requires few and sometimes no parameters. However, it is still prone to inferior solutions due to the limited exploration and exploitation ability.

There are two major approaches to hybridize the VNS with other metaheuristics to improve its performance: hybridizing with a local base metaheuristic and hybridizing with a population based metaheuristic. The first idea is to embed SA into VNS, so that it is replaced with local search, whereas SA in hybrid VNS addresses how to get out of large valleys. Besides, SA acts as the local search method, because it is good at searching the neighborhood of a solution. The three neighborhoods employed are swap, insertion, and inversion.

As one of the most well-known population based methods, genetic algorithm (GA) shows robust performance with various problems. Usually, GA has been proven to be very good at shuffling the solution space or global exploration ability but fail to intensify the search towards promising regions. Nevertheless, GA usually takes more computing efforts to locate the optimal in the region of convergence [27], owing to the lack of local search ability. Therefore, hybridization with local search methods may overcome this weakness and lead to powerful search schemes. So, the second idea is to embed VNS as a local search into GA and may be a likely choice to consider the hybridization of them. In GA, VNS is applied as a local search to a subset of offspring generated by one-point

TABLE 1: Test problems characteristics.

Parameters	Levels	Count
Number of jobs (N)	10, 20, 30, 50, 75, 100, 125, 150, 175 and 200	10
Processing time of jobs (P)	Uniform distributions [1, 10], [1, 20]	2
Size of jobs (S)	Uniform distributions [1, 10], [2, 4], [4, 8]	3
Tardiness cost (T)	[5, 8]	1
$d_{j,u}$	Uniform distributions (0.7 × BP, BP)	1
$d_{j,l}$	Uniform distributions (0.6 × $d_{j,u}$, 0.8 × $d_{j,u}$)	1
Cap	10	1
Total number of problem instances		60

crossover and swap mutation operator to search for better solutions.

4. Computational Experiments

4.1. Instances. To compare the proposed algorithms, some test problems are needed. In this regard, we generate the required data that can affect the performance of the algorithms including the number of jobs (n), range of processing time of jobs (p_j), size of jobs (s_j), tardiness costs (β_i), and due date of jobs (d_j). The crisp due dates in Tavakkoli-Moghaddam et al. [28] test problems are generated from a uniform distribution. We use such procedure with some modifications to adapt the procedure for our problem as follows:

$$\bar{P} = \frac{\sum_{j=1}^n P_j}{n},$$

$$\bar{B} = \frac{\sum_{j=1}^n s_j}{0.8 \times \text{cap}}, \tag{25}$$

$$BP = \bar{B} \times \bar{P}.$$

After generating the BP , the $d_{j,l}$, and $d_{j,u}$ are generated as explained in Table 1.

4.2. Parameter Setting. Because of the dependency of metaheuristic algorithms on the correct selection of parameters and operators, we study the behavior of different parameters of proposed algorithms. The parameters of proposed algorithms are as follows: initial temperature (T_0), number of neighborhood search (n_{\max}), reduction ratio of temperature (α) population size ($popsiz$), crossover percentage (p_c), and mutation probability (p_m). Levels of these factors are illustrated in Table 2.

In order to be fair, the stopping criterion for all algorithms is equal to $6 \times n$ milliseconds. This criterion is sensitive to the problem size. Using this stopping criterion, searching time increases according to the rise in number of jobs. To yield more reliable information and due to having stochastic nature

TABLE 2: Factors and their levels.

Parameters	SA, VNS and VNS-SA			Parameters	GA and GA-VNS	
	SA levels	VNS levels	VNS-SA levels		GA levels	GA-VNS levels
T_0	A(1)—300	—	A(1)—200	$popsize$	A(1)—45	A(1)—30
	A(2)—350	—	A(2)—250		A(2)—50	A(2)—35
	A(3)—400	—	A(3)—300		A(3)—60	A(3)—40
n_{max}	B(1)—600	A(1)—400	B(1)—400	p_c	B(1)—80%	B(1)—80%
	B(2)—650	A(2)—450	B(2)—450		B(2)—85%	B(2)—85%
	B(3)—700	A(3)—500	B(3)—500		B(3)—90%	B(3)—90%
α	C(1)—0.91	—	C(1)—0.89	p_m	C(1)—0.1	C(1)—0.1
	C(2)—0.92	—	C(2)—0.9		C(2)—0.15	C(2)—0.15
	C(3)—0.93	—	C(3)—0.91		C(3)—0.2	C(3)—0.2
				n_{max}	D(1)—300	D(1)—300
					D(2)—350	D(2)—350
					D(3)—400	D(3)—400

of algorithms, we tackle each test problem ten times. Because the scale of objective functions in each instance is different, they cannot be used directly. To solve this problem, the relative percentage deviation (RPD) is used for each instance. The RPD is obtained by the following formula:

$$RPD = \frac{Alg_{sol} - Min_{sol}}{Min_{sol}} \times 100, \quad (26)$$

where Alg_{sol} and Min_{sol} are the obtained objective value and minimum objective value found from both proposed algorithms for each instance, respectively. So, we use the RPD measure in the proposed algorithms.

After obtaining the results of the test problems, the results are transformed into RPD measures. The RPD measures are averaged and their value is depicted in Figures 2–6. In SA, better robustness happens when parameters T_0 , n_{max} and α are 350, 650, and 0.92, respectively, as depicted in Figure 2. In Figure 3, the RPD measure for the single parameter of VNS (n_{max}) is depicted and the second level or 450 is the best. Also, for hybrid VNS, as illustrated in Figure 4, T_0 , n_{max} , and α are defined as 250, 450, and 0.9. In conformity with Figure 5, best magnitude for $popsize$, p_c , and p_m in GA are 50, 85%, and 15. Besides, in accordance with Figure 6, in the proposed GA-VNS, best quantity for $popsize$, p_c , p_m , and n_{max} are 35, 85%, 15, and 350, respectively.

4.3. Experimental Results. In this section, we present and compare the results of EDDL, EDDU, SA, VNS, GA, VNS-SA, and GA-VNS with the EDD dispatching rule as a well-known heuristic algorithm related to the due date. As mentioned above, we have 60 problem instances, in which each one includes 10 performed replications to achieve the more reliable results. Table 3 demonstrated the results obtained from EDD, EDDL, and EDDU, in which the first and fourth columns represent the data sets characteristics and the remaining columns show the results on instances.

According to Table 3, among heuristics, EDDL has the worst results, and it can be concluded that EDDU is better than EDD. In order to analyze the interaction between quality

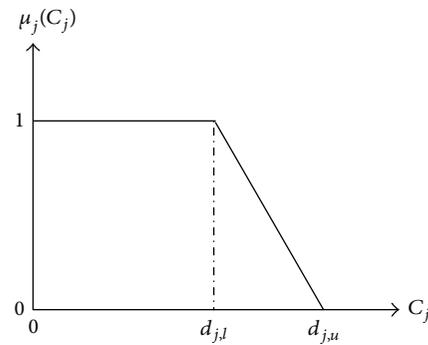


FIGURE 1: Membership function.

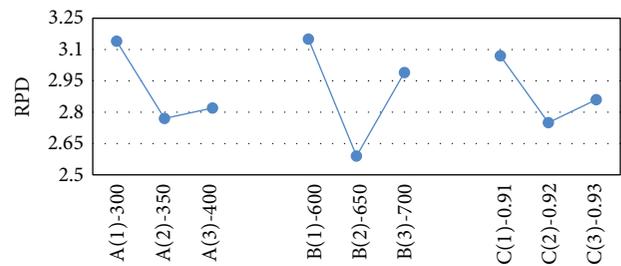


FIGURE 2: Mean RPD plot for each level of the factors in SA.

of the algorithms and different problem sizes more concisely, the RPD results are calculated for test problems and averaged for each problem size. The average RPDs obtained by each algorithm are shown in Figures 7 and 8. In these figures, each point represents the average results obtained from six test problems considered in each size of problems with ten replications in each algorithm.

It is noticeable that with increasing the problem size, gradually the RPDs of the proposed EDD, EDDL, and EDDU decrease. In spite of decreasing the RPDs, they are not capable to be complete even in the four last sizes.

As it can be seen, the GA-VNS keeps its robust performance in all ranges of problem sizes. On the contrary to SA

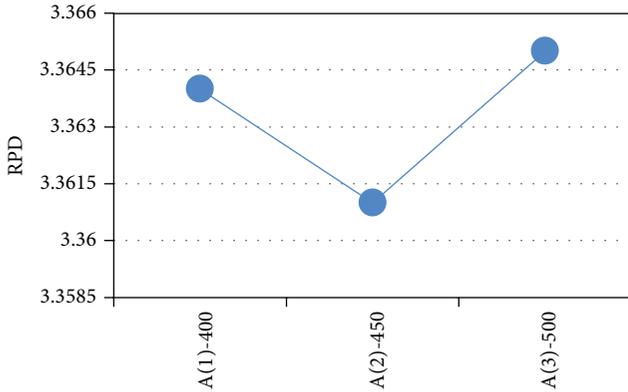


FIGURE 3: Mean RPD plot for each level of the factors in the VNS.

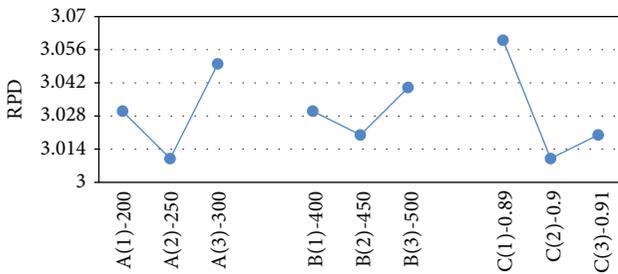


FIGURE 4: Mean RPD ratio plot for each level of the factors in VNS-SA.

and VNS, both VNS-SA and GA have good results. VNS-SA has better performance in 10 j and 30 j; however, in the last six problem sizes, GA outperforms it. In the first problem size, SA yields the best result, but with increasing the problem size, gradually its RPDs increase. In the first four problem sizes, VNS does not show a good performance, but with increasing the problem size, its RPDs decrease, while in latter sizes it outperforms SA.

From Figures 7 and 8, there is no significant difference between proposed EDD and EDDU or SA and VNS. So, we perform an analysis of variance (ANOVA) to accurately analyze the results among them. The means plot and LSD intervals (at the 95% confidence level) for the presented algorithms are shown in Figures 9 and 10. According to the results, the average RPD obtained by the proposed EDD, EDDL, and EDDU are 128.04, 141, and 132.33 respectively. So, EDD is better than EDDU and EDDL.

Also, the average RPD obtained by the proposed GA-VNS is 0.22, while for GA, VNS-SA, SA, and VNS are 0.37, 0.41, 0.6, and 0.65, respectively. As is evident, GA-VNS has outperformed other algorithm. As it can be seen, between GA and VNS-SA, and also between SA and VNS, there is not a significant difference. However, they failed to statistically overcome each other. However, based on the results, we conclude that the proposed GA-VNS can be used to effectively solve the problem.

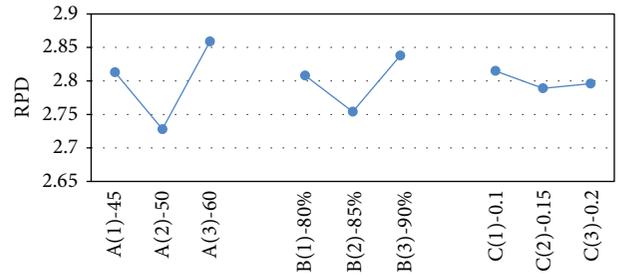


FIGURE 5: Mean RPD ratio plot for each level of the factors in GA.

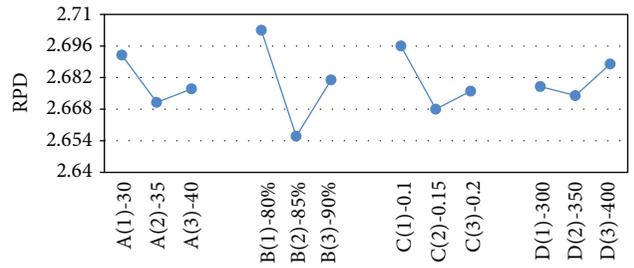


FIGURE 6: Mean RPD ratio plot for each level of the factors in GA-VNS.

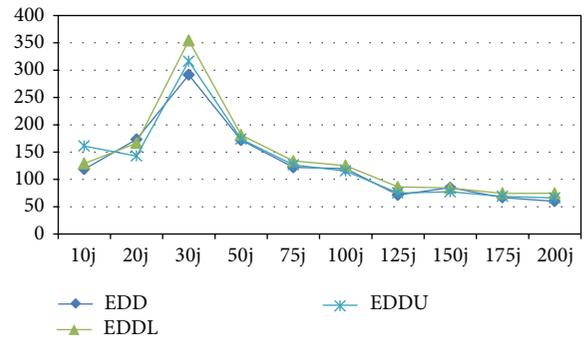


FIGURE 7: Means plot for the interaction among heuristic algorithms.

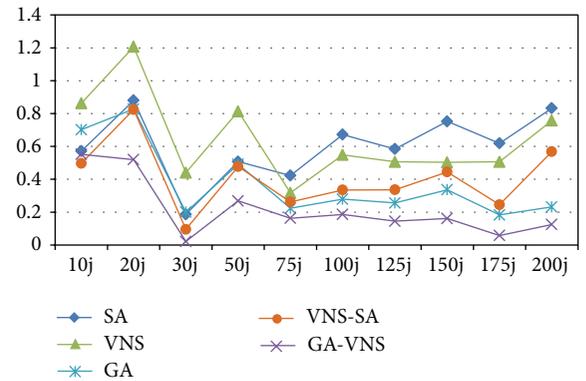


FIGURE 8: Means plot for the interaction among metaheuristic algorithms.

TABLE 3: Results of EDD, EDDL, and EDDU on test problems.

Problem	EDD	EDDL	EDDU
10jp1s1	113.9	101.15	113.9
10jp1s2	140.49	189.19	140.49
10jp1s3	123.16	91.03	132.06
10jp2s1	119.33	130.42	185.23
10jp2s2	113.42	114.74	113.42
10jp2s3	155.56	166.44	185.35
20jp1s1	203.7	208.63	189.57
20jp1s2	196.16	190.16	175.72
20jp1s3	256.24	204.22	203.58
20jp2s1	120.96	134.36	122.45
20jp2s2	214.69	193.68	162.22
20jp2s3	153.54	172.27	153.37
30jp1s1	195.61	193.39	198.37
30jp1s2	252.3	267.01	252.26
30jp1s3	488.24	772.72	569.51
30jp2s1	191.91	234.04	197.27
30jp2s2	351.17	362.43	379.27
30jp2s3	396.51	400.85	410.41
50jp1s1	539.52	686.09	476.37
50jp1s2	424.3	443.42	398.52
50jp1s3	708.06	682.57	701.39
50jp2s1	559.46	591.87	665.65
50jp2s2	410.91	423.84	412.4
50jp2s3	527.77	390.2	565.44
75jp1s1	761.27	707.47	701.15
75jp1s2	594.23	635.03	515.88
75jp1s3	682.1	774.99	716.68
75jp2s1	896.54	1085.56	1117.63
75jp2s2	517.24	507.16	507.64
75jp2s3	872.84	852.19	917.32
100jp1s1	573.12	631.37	655.84
100jp1s2	874.41	932.05	864.1
100jp1s3	1015.52	1032.15	1052.11
100jp2s1	991.97	1047	946.9
100jp2s2	810.87	716.69	636.58
100jp2s3	1365.03	1402.12	1342.32
125jp1s1	1263.82	1602.84	1236.04
125jp1s2	978.41	910.7	989.08
125jp1s3	1090.05	1125.03	1088.53
125jp2s1	968.01	1303.24	1148.23
125jp2s2	962.34	942.8	947.46
125jp2s3	1261.83	1250.44	1254.45
150jp1s1	1523.47	1473.88	1199.04
150jp1s2	1185.04	1079	1417.27
150jp1s3	1654.88	1730.38	1579.02
150jp2s1	992.93	992.75	897.83
150jp2s2	1167.75	1279.56	1143.29
150jp2s3	1727.18	1684.63	1676.66
175jp1s1	1150.47	1261.93	1025.12
175jp1s2	1303.52	1414.09	1335.53

TABLE 3: Continued.

Problem	EDD	EDDL	EDDU
175jp1s3	1888.85	1805.08	2013.34
175jp2s1	1623.61	1761.51	1876.28
175jp2s2	1408.78	1437.47	1406.21
175jp2s3	1651.22	1699.95	1520.91
200jp1s1	1198.46	1142.2	1160.17
200jp1s2	1556.72	1828.91	1724.02
200jp1s3	2266.57	2261.79	1935.25
200jp2s1	1486.81	1935.52	1912.32
200jp2s2	1708.57	1774.06	1731.21
200jp2s3	1892.47	2006.77	1885.73

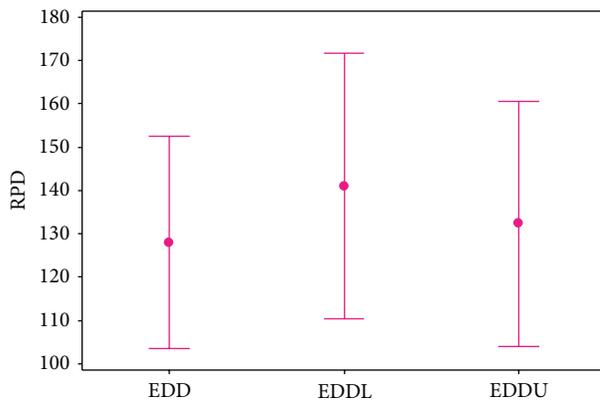


FIGURE 9: Means plot and LSD intervals for proposed heuristics.

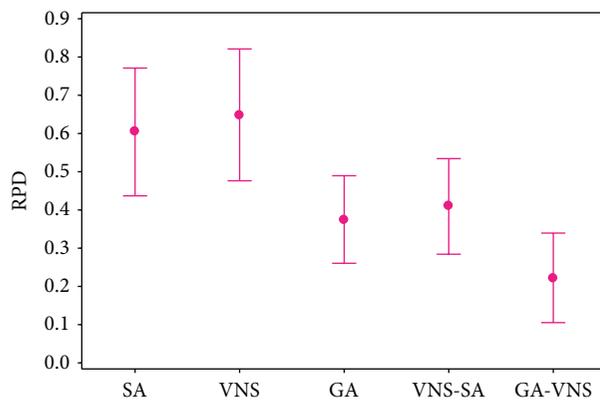


FIGURE 10: Means plot and LSD intervals for proposed metaheuristics.

5. Conclusions and Future Research Directions

In this paper, we discussed the single batch-processing machine (SBPM) scheduling problem in the presence of fuzzy due date to minimize the total weighted tardiness. We developed a mixed integer linear programming model with the objective functions of the total satisfaction or dissatisfaction degree. To solve this model, three heuristics (EDD, EDDL, and EDDU), three metaheuristics (GA, VNS,

and SA), and two hybrid metaheuristics (GA-VNS and VNS-SA) are developed. Also, a plan was developed and utilized to generate test problems in a fuzzy environment. To enhance the performance of the proposed method, the experimental design method was used by setting their parameters. The computational results showed that the hybrid GA-VNS were robust and superior to other proposed algorithms. As a future work, total weighted earliness tardiness can be considered as the objective function and the same proposed algorithms can be developed for it. Another direction is to work on other algorithms, such as Cuckoo Optimization Algorithm [29], Honey Bees Optimization [30], Differential Evolution [31], Cuckoo Search [32], and Firefly Algorithm [25, 26].

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] R. Uzsoy, C. Lee, and L. Martin-Vega, "A review of production planning and scheduling models in the semiconductor industry. Part I. System characteristics, performance evaluation and production planning," *IIE Transactions*, vol. 24, pp. 47-60, 1992.
- [2] R. Uzsoy, C. Y. Lee, and L. A. Martin-Vega, "Review of production planning and scheduling models in the semiconductor industry part II: shop-floor control," *IIE Transactions*, vol. 26, no. 5, pp. 44-55, 1994.
- [3] Y. Ikura and M. Gimple, "Efficient scheduling algorithms for a single batch processing machine," *Operations Research Letters*, vol. 5, no. 2, pp. 61-65, 1986.
- [4] C. Y. Lee, R. Uzsoy, and L. A. Martin-Vega, "Efficient algorithms for scheduling semiconductor burn-in operations," *Operations Research*, vol. 40, no. 4, pp. 764-775, 1992.
- [5] C. N. Potts and M. Y. Kovalyov, "Scheduling with batching: a review," *European Journal of Operational Research*, vol. 120, no. 2, pp. 228-249, 2000.
- [6] M. Mathirajan and A. I. Sivakumar, "A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor," *International Journal of Advanced Manufacturing Technology*, vol. 29, no. 9-10, pp. 990-1001, 2006.

- [7] C. S. Wang and R. Uzsoy, "A genetic algorithm to minimize maximum lateness on a batch processing machine," *Computers and Operations Research*, vol. 29, no. 12, pp. 1621–1640, 2002.
- [8] S. Melouk, P. Damodaran, and P.-Y. Chang, "Minimizing makespan for single machine batch processing with non-identical job sizes using simulated annealing," *International Journal of Production Economics*, vol. 87, no. 2, pp. 141–147, 2004.
- [9] S. G. Koh, P. H. Koo, D. C. Kim, and W. S. Hur, "Scheduling a single batch processing machine with arbitrary job sizes and incompatible job families," *International Journal of Production Economics*, vol. 98, no. 1, pp. 81–96, 2005.
- [10] M. Sevaux and S. Dauzère-Pérès, "Genetic algorithms to minimize the weighted number of late jobs on a single machine," *European Journal of Operational Research*, vol. 151, no. 2, pp. 296–306, 2003.
- [11] A. Husseinzadeh Kashan, B. Karimi, and F. Jolai, "Effective hybrid genetic algorithm for minimizing makespan on a single-batch-processing machine with nonidentical job sizes," *International Journal of Production Research*, vol. 44, pp. 2337–2360, 2006.
- [12] P. Damodaran, P. Kumar Manjeshwar, and K. Srihari, "Minimizing makespan on a batch-processing machine with non-identical job sizes using genetic algorithms," *International Journal of Production Economics*, vol. 103, no. 2, pp. 882–891, 2006.
- [13] L. Mönch, R. Unbehaun, and Y. I. Choung, "Minimizing earliness-tardiness on a single burn-in oven with a common due date and maximum allowable tardiness constraint," *OR Spectrum*, vol. 28, no. 2, pp. 177–198, 2006.
- [14] F. D. Chou, P. C. Chang, and H. M. Wang, "A hybrid genetic algorithm to minimize makespan for the single batch machine dynamic scheduling problem," *International Journal of Advanced Manufacturing Technology*, vol. 31, no. 3-4, pp. 350–359, 2006.
- [15] H. M. Wang, P. C. Chang, and F. D. Chou, "A hybrid forward/backward approach for single batch scheduling problems with non-identical job sizes," *Journal of the Chinese Institute of Industrial Engineers*, vol. 24, no. 3, pp. 191–199, 2007.
- [16] A. H. Kashan and B. Karimi, "Scheduling a single batch-processing machine with arbitrary job sizes and incompatible job families: an ant colony framework," *Journal of the Operational Research Society*, vol. 59, no. 9, pp. 1269–1280, 2008.
- [17] F. D. Chou and H. M. Wang, "Scheduling for a single semiconductor batch-processing machine to minimize total weighted tardiness," *Journal of the Chinese Institute of Industrial Engineers*, vol. 25, no. 2, pp. 136–147, 2008.
- [18] M. Mathirajan, V. Bhargav, and V. Ramachandran, "Minimizing total weighted tardiness on a batch-processing machine with non-agreeable release times and due dates," *International Journal of Advanced Manufacturing Technology*, vol. 48, no. 9-12, pp. 1133–1148, 2010.
- [19] H. M. Wang, "Solving single batch-processing machine problems using an iterated heuristic," *International Journal of Production Research*, vol. 49, no. 14, pp. 4245–4261, 2011.
- [20] A. Husseinzadeh Kashan, B. Karimi, and F. Jolai, "An effective hybrid multi-objective genetic algorithm for bi-criteria scheduling on a single batch processing machine with non-identical job sizes," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 6, pp. 911–922, 2010.
- [21] H. Ishii, M. Tada, and T. Masuda, "Two scheduling problems with fuzzy due-dates," *Fuzzy Sets and Systems*, vol. 46, no. 3, pp. 339–347, 1992.
- [22] K. K. Harikrishnan and H. Ishii, "Single machine batch scheduling problem with resource dependent setup and processing time in the presence of fuzzy due date," *Fuzzy Optimization and Decision Making*, vol. 4, no. 2, pp. 141–147, 2005.
- [23] A. D. Yimer and K. Demirli, "Fuzzy scheduling of job orders in a two-stage flowshop with batch-processing machines," *International Journal of Approximate Reasoning*, vol. 50, no. 1, pp. 117–137, 2009.
- [24] B. Cheng, K. Li, and B. Chen, "Scheduling a single batch-processing machine with non-identical job sizes in fuzzy environment using an improved ant colony optimization," *Journal of Manufacturing Systems*, vol. 29, no. 1, pp. 29–34, 2010.
- [25] Y. Zhang, S. Wang, G. Ji, and Z. Dong, "Genetic pattern search and its application to brain image classification," *Mathematical Problems in Engineering*, vol. 2013, Article ID 580876, 8 pages, 2013.
- [26] Y. Zhang, L. Wu, and S. Wang, "Solving two-dimensional HP model by firefly algorithm and simplified energy function," *The Scientific World Journal*, vol. 2013, Article ID 398141, 9 pages, 2013.
- [27] A. S. Wu, H. Yu, S. Jin, K.-C. Lin, and G. Schiavone, "An incremental genetic algorithm approach to multiprocessor scheduling," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 9, pp. 824–834, 2004.
- [28] R. Tavakkoli-Moghaddam, A. Rahimi-Vahed, and A. H. Mirzaei, "A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: weighted mean completion time and weighted mean tardiness," *Information Sciences*, vol. 177, no. 22, pp. 5072–5090, 2007.
- [29] S. Balochian and E. Ebrahimi, "Parameter optimization via cuckoo optimization algorithm of fuzzy controller for liquid level control," *Journal of Engineering*, vol. 2013, Article ID 982354, 7 pages, 2013.
- [30] Y. Celik and E. Ulker, "An improved marriage in honey bees optimization algorithm for single objective unconstrained optimization," *The Scientific World Journal*, vol. 2013, Article ID 370172, 11 pages, 2013.
- [31] T. J. Choi, C. W. Ahn, and J. An, "An adaptive cauchy differential evolution algorithm for global numerical optimization," *The Scientific World Journal*, vol. 2013, Article ID 969734, 12 pages, 2013.
- [32] Y. Zhou and H. Zheng, "A novel complex valued cuckoo search algorithm," *The Scientific World Journal*, vol. 2013, Article ID 597803, 6 pages, 2013.




Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

