Scientia Iranica E (2014) 21(3), 1007–1020

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
www.scientiairanica.com

An enhanced invasive weed optimization for makespan minimization in a flexible flowshop scheduling problem

F. Jolai[a,*], R. Tavakkoli-Moghaddam[a], M. Rabiee[b] and E. Gheisariha[c]

a. *School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, P.O. Box 11155/4563, Iran.*
b. *Department of Industrial Engineering, K. N. Toosi University of Technology, Tehran, Iran.*
c. *Department of Industrial Engineering, University of Tehran, Kish Int'l Campus, Iran.*

Received 11 Aguste 2012; received in revised form 15 May 2013; accepted 14 January 2014

KEYWORDS
IWO;
Flexible flow shop;
Response surface methodology;
Rework;
Transportation time;
Anticipatory sequence dependent setup time.

Abstract. In previous investigations in the field of flexible flow shop scheduling problems, the rework probability for operations was ignored. As these kinds of problems are NP-hard, we present an Enhanced Invasive Weed Optimization (EIWO) algorithm in order to solve the addressed problem with probable rework times, transportation times with a conveyor between two subsequent stages, different ready times and anticipatory sequence dependent setup times. The optimization criterion is to minimize makespan. Although Invasive Weed Optimization (IWO) is an efficient meta-heuristic algorithm and has been used by many researchers recently, to increase the capability of IWO, we added a mutation operation to enhance the exploration in order to prevent sticking in local optimum. In addition, an affinity function is embedded to obstruct premature convergence. With these changes, we balance the exploration and exploitation of IWO. Since the performance of our proposed algorithm depends on parameters values, we apply the popular design of an experimental methodology, called the Response Surface Method (RSM). To evaluate the proposed algorithm, first, some random test problems are generated and compared with three benchmark algorithms. The related results are analyzed by statistical tools. The experimental results and statistical analyses demonstrate that the proposed EIWO is effective for the problem.

1. Introduction

This research is concerned with scheduling problems in Flexible Flow Line (FFL) manufacturing systems with ready time, Anticipatory Sequence Dependent Setup Times (ASDST), transportation with a conveyor between two consecutive stages and probable rework for each job operation. An FFL is also called a Flexible Flow Shop (FFS), a Hybrid Flow Shop (HFS), or a Flow Shop with Multiple Processors (FSMP). The flexible flow shop is a generalized case of the basic flow shop. It sequences n jobs in a flow shop, where more than one identical machine at one or more than one stage is allowable. The flexible flow line environment is fairly common and appears in flow industries such as steel, petroleum, chemical processing, packaging [1], automobile assembly [2] and ceramic tile manufacturing [3]. A brief history of work done on this problem using various solution methods and different constraints and assumptions includes the following.

Arthanari and Ramamurthy [4] and Salvador [5] were among the first to define the flexible flow shop problem. They proposed a branch-and-bound method to tackle the problem. Such a method is an exact

*. *Corresponding author. Tel.: +98 21 88021067; Fax: +98 21 88013102*
E-mail addresses: fjolai@ut.ac.ir (F. Jolai); tavakoli@ut.ac.ir (R.Tavakkoli-Moghaddam); meysam_rabiee@basu.ac.ir (M. Rabiee); Gheisariha.e@gmail.com (E. Gheisariha)

solution technique, which guarantees optimal solutions. However, the exact algorithm presented can only be applied to very small instances. Other exact approaches for the multi-stage flexible flow shop problem were proposed by many authors, e.g. branch-and-bound algorithms were given by Brah and Hunsucker [6] and Moursli and Pochet [7]. Gupta [8] proposed heuristic techniques for a simplified flexible flow shop makespan problem with two stages and only one machine at stage two. Azizoglu et al. [9] presented a branch and bound algorithm to solve the flexible flow shop where the solutions are not restricted to permutation schedules. Azizoglu et al. [9] a branching scheme and tested the proposed algorithm on two data sets, one with small and the other with large processing times. Sawik [10] modeled a flexible flow line with blocking and reentry. This model was later improved by Sawik [11]. Lee and Kim [12] proposed a branch and bound solution approach to minimize total tardiness. Problem instances of up to 15 jobs were illustrated to be solvable in reasonable times. Bolat et al. [13] investigated the two-stage hybrid flow shop scheduling problem with a single machine at the first stage and two identical machines at the second stage. They solved the mentioned problem with branch and bound, heuristics, and genetic algorithms. Sawik [14] proposed more mathematical models for a flexible flow line with tardiness related criteria. Haouari et al. [15] solved a two-stage regular HFS (unconstrained number of machines in stages 1 and 2) to minimize makespan criterion with a very effective branch and bound method that produces optimal solutions for problems up to 1000 jobs in size. However, they showed, in some cases, that the observed average gap reached more than 4%. The HFS is modelled as a resource constrained multi-project scheduling problem with setup times [16].

Since branch & bound algorithms are usually time consuming for large scale problems (since the problem is NP-hard), many researchers used heuristic approaches. Narasimhan and Panwalkar [17] considered a real-life FFL with one machine at stage 1 and two machines at stage 2. The CMD (Cumulative Minimum Deviation) rule was suggested to reduce the sum of machine idle time and in-process job waiting time. Later, Narasimhan and Mangiameli [18] proposed the GCMD (Generalized Cumulative Minimum Deviation) rule, which is an extension of the CMD rule, for the FFL with five criteria. Ding and Kittichartphayak [19] developed three heuristics for scheduling in FFL $(Pm1, Pm2, ..., PmS)||C_{\max}$. The computational results showed that one of their heuristics, called the Combined Approach, is the best and can solve problem sets with up to 8 jobs with an average error less than 3% of the optimal solutions. Sriskandarajah and Sethi [20] developed simple heuristic algorithms for the two-stage flexible flow shop problem. They discussed the worst and average case performance of algorithms in order to find minimum makespan schedules. Their solutions are based on Johnson's rule. Braglia and Petroni [21] applied data development analysis from the results of Kadipasaoglu et al. [22] as a means to obtain accurate information about the performance of dispatching rules. Kurz and Askin [23] contemplate dispatching rules for the hybrid flow shop with Sequence Dependent Setup Times (SDST). They studied three types of heuristics based on some simple greedy algorithms, insertion heuristics and adaptations of Johnson's rule. A recent study about the dispatching rule in hybrid flow shop scheduling problems was carried out by Lee [24].

Recently, meta-heuristics have become quite popular compared to other approximate, exact or heuristic methods for solving complex combinatorial optimization problems, such as job shop, flow shop scheduling problems and many other hard problems [25-36]. Meta-heuristics have been highly successful in finding optimal or near-optimal solutions for any practical scheduling and sequencing problems, particularly in FFL. Haouari and M'Hallah [37] used two two-phased heuristics based on simulated annealing and Tabu search to solve the two-stage flexible flow shop with parallel machines at each stage. To construct an initial solution, the most work remaining rule was used to create a priority list. The objective was to minimize makespan. Tabu search performed just a little better than simulated annealing. Riane et al. [38] presented a simulated annealing algorithm to solve the flexible flow shop problem in which only permutation sequences are considered. The paper emphasized the importance of temperature reduction; spending too much time at a high temperature wastes time and decreasing the temperature too quickly will limit the search to local optima. Negenman [39] tested a variable depth search method combined with three simulated annealing heuristics and three Tabu search heuristics to solve the flexible flow shop. The combination of a variable depth search with the Tabu search performed the best. Kurz and Askin [40] presented an Integer Programming (IP) model for HFS with SDST. Because of the difficulty in solving IP models directly, they developed a Random Keys Genetic Algorithm (RKGA) to tackle the studied problem against other heuristic rules. Zandieh et al. [41] proposed an immune algorithm, and compared it favourably against the RKGA of Kurz and Askin [40]. The hybrid flow shop with unrelated machines and SDST was considered by Ruiz and Maroto [3]. To tackle the problem, the authors introduced a genetic algorithm. Naderi et al. [42] studied a realistic case of flow shops with parallel machine setup times, where the objective is makespan minimization. They also assumed that each job might not need to visit all stages. They introduced a heuristic, in the form of a dynamic dispatching rule, and a meta-heuristic, based on iterated local search. Behnamian and Zandieh [43] proposed a discrete colonial competitive algorithm to solve a hybrid flow shop scheduling problem with limited waiting time between two subsequent stages and sequence dependent setup times. To understand more literature review about the hybrid flow shop scheduling problem, readers can refer to two state-of-the-art papers in this area [44,45].

Among previous investigations in HFS, there are two studies which are worthy of review because of their similarity to our study. Naderi et al. [46] proposed an improved version of simulated annealing to solve the hybrid flow shop scheduling problem, Anticipatory Sequence Dependent Setup Times (ASDST) and transportation time between two consecutive stages, to minimize total completion time and total tardiness. They showed that their algorithm is better than the other approaches. Naderi et al. [47] proposed a mixed integer programming model for small size problems and also a method, called the Electromagnetism Algorithm (EMA), to tackle scheduling the FFSSP, with anticipatory sequence dependent setup times and transportation time, with the aim of minimizing total weighted tardiness. In these two later reviewed studies, for travelling between the stages, some transporters, like the Automatic Guided Vehicle (AGV), are considered. This type of transportation increases the waiting time between two consecutive stages because the next job can be processed when AGV comes back to the loading centre/point. In other words, for example, if two jobs were ready to be processed simultaneously, the job with lower priority should wait until the loading time, the travelling time and the unloading time finish.

Regarding the discussed literature and to the best of our knowledge, in FFSSP with ASDST, transportation time by conveyor, ready time and probable rework has not been investigated simultaneously. In order to solve the addressed problem, we developed a novel meta-heuristic algorithm which is based on invasive weed optimization. The rest of this paper is organized as follows. Section 2 elaborates on the problem considered in this paper, followed by assumptions. Section 3 presents the proposed meta-heuristic method. Section 4 describes generation of the test data, parameter tuning and analysis of computational experiments. Finally, section 5 summarizes the major findings of this paper and proposes some promising directions for future research in this area.

2. Problem definition

Flexible flow shop is a kind of machine scheduling problem. There are n jobs to be processed. Job j ($j = 1, 2, ..., n$) has to be processed at each stage i ($i = 1, 2, ..., s$) in series. There are m_i identical parallel machines at stage i. The processing times of job j at stage i are denoted by $p_{i,j}$. The sequence-dependent setup time between job j and job k at stage i is depicted by s^i_{jk}. For transportation of job j between stage i and l, three terms are defined: Loading time, travelling time, and unloading time which are denoted by $lt^j_{i,l}$, $tt_{i,l}$ and $ut^j_{i,l}$, respectively. After processing of each job at each stage on each machine, an inspection procedure is considered. Inspection time is a segment of the processing time. After inspection, with predetermined probability (rp^i_j), each job may need to rework the procedure which is defined by rework time (rt^j_i). For better understanding, the problem studied in this paper is depicted in Figure 1.

The problem mentioned in this study can be categorized under three sub-problems. The first is to find the best sequence for jobs. The second is to define how the jobs are assigned to machines for processing. The strategy applied for this purpose is First Available Machine (FAM), used in the literature [48-50]. The third is to design a strategy for jobs in the transportation step. For this issue, two rules are considered. One is: First In First Out (FIFO), and another is the priority of jobs in sequence. These rules are discussed in Subsection 4.2.

Considering the scheduling problems, a certain number of assumptions have to be made to proceed

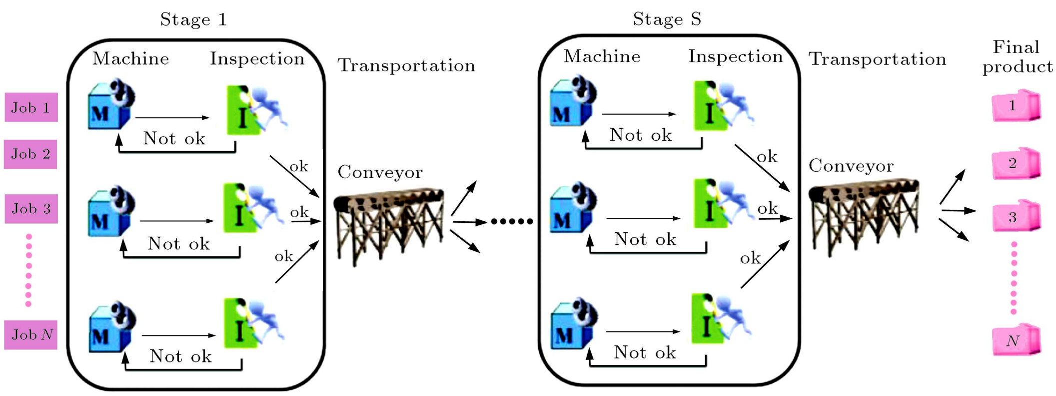

Figure 1. Schematic of the problem studied in this paper.

towards the solution methodologies. Some general and exclusive assumptions about the considered system in this work are stated as follows:

- No preemption: A job already started on a station must be completed before another job can start on that station, i.e. a new coming job cannot preempt the already under-process job and, therefore, must wait for scheduling until the previous job finishes.
- Each job should be processed once.
- The processing times of the jobs are independent of the schedule and may be different from one stage to another.
- No machine can process more than one job simultaneously.
- Machines may be idle waiting for the next job to be released from any previous machine in the production line.
- Machines never breakdown and are available throughout the scheduling period.
- The system has no buffers between stages (zero buffer system).
- Each job should go through all production stages in the same order as all other jobs.
- Processing time of each job is equal on all processors at each stage, and each job can be processed on any processor in the stage (processors are identical).
- Each stage contains at least one processor. In case of more than one processor, the processors are considered identical.
- Transportation time is job independent and is related to the distance between two consecutive stages.
- It is assumed that there is enough labor besides the conveyors. In other words, there is no delay time for jobs due to shortage of labor.

The standard flow shop with more than two machines, with the objective of minimizing makespan, is considered to be NP-Hard in the strong sense [51]. This problem is considerably more complex because it adds parallel machines at each stage, ready time, probable rework, transportation time and anticipatory sequence dependent setup times. As mentioned above, flexible flow-shop problems are NP-Hard. No algorithms except exhaustive search have ever been provided for finding optimal solutions. In order to solve this weighty matter, we developed an enhanced version of invasive weed optimization. The detailed procedure of our proposed algorithm is described in the next section.

Figure 2. Solution representation in EIWO.

Figure 3. Decoded solution (sequence).

3. Enhanced invasive weed optimization

Invasive Weed Optimization is described by Mehrabian and Lucas [52]. The IWO algorithm is a numerical stochastic search algorithm and is a population-based intelligence algorithm which mimics the colonizing behavior of weeds in finding suitable places for growth. During recent years, many researchers have claimed that this algorithm has been very efficient in solving their studied problems [53-56].

3.1. Population initialization

A bounded number of weeds, called n_{pop}, are initialized randomly by a uniform distribution in a range between zero and one (see Figure 2). Then, these values are transformed to sequences for fitness evaluation (see Figure 3). This way, we have a primary population, which is called pop1 (i.e. pop1 $= n_{\text{pop}}$).

3.2. Reproduction

Reproduction in IWO is the same as reproduction of chromosomes in GA. Due to the eligibility of the population, each solution that belongs to the population is allowed to produce seeds within a specified region centered at its own position. The number of seeds produced around each solution depends on its relative fitness in the population, with respect to the best and worst fitness. The number of seeds produced by any weed varies linearly from maximum possible seed ($S_{\max}$) (for best solution) to minimum possible seeds ($S_{\min}$) (for worst solution). The population generated in this step is denoted by pop. The number of seeds (S_i) around each solution is computed by Eq. (1).

$$S_i = S_{\min} + (n_{\text{pop}} - \text{rank}_i) \times S_{\max}, \qquad (1)$$

where n_{pop} is the highest number of weeds in the colony, and rank_i is the rank of the ith weed in the colony. The schematical seed reproduction procedure is illustrated in Figure 4.

For better clarification, this concept is shown in Figure 5. In this figure, it is assumed that weed1 and weed5 are the best and worst weeds between five given weeds. So, the number of seeds around weed1 is equal to $S_{\max}$ and the number of seeds around weed5 is equal to $S_{\min}$.

3.3. Spatial distribution

To apply the seed reproduction step, a random value array in the size of the job number is generated by

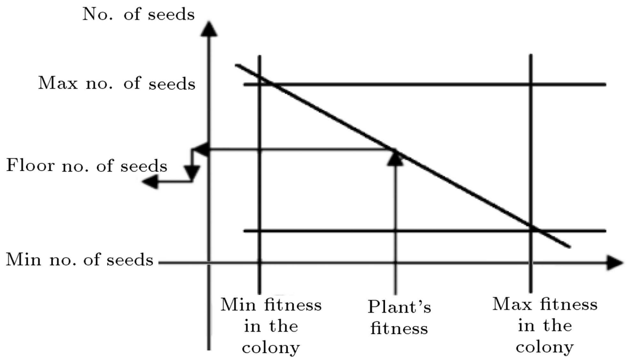

Figure 4. Seeds reproduction procedure [52].

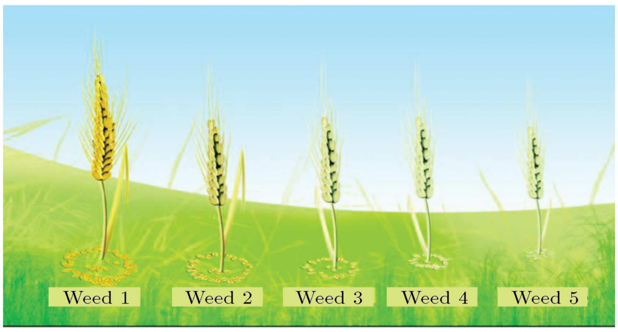

Figure 5. Schematic of seed reproduction procedure for a problem with 5 weeds.

a normal distribution, with mean equal to zero and variance ∂^2. Then, the generated values are added to the previous values of the related weed and, similar to the first step, the real values are transformed into a sequence for fitness evaluation. This step ensures that the generated seeds will be produced around the parent weed, leading to a local search around each plant. To achieve this purpose, at the beginning of the algorithm, the value of the standard deviation of each weed is greater than its value when the algorithm reaches stopping criteria. If σ_{init} and σ_{final} are the initial and final standard deviations, we can use Eq. (2) to specify iteration:

$$\sigma_{\text{iter}} = \frac{(\text{iter}_{\max} - \text{iter})^{\text{pow}}}{(\text{iter}_{\max})^{\text{pow}}}(\sigma_{\text{init}} - \sigma_{\text{final}}) + \sigma_{\text{final}}, \quad (2)$$

where iter is the current iteration number and $\text{iter}_{\max}$ equals the maximum number of iterations allowed. An illustration of decreased standard deviation is shown in Figure 6. As seen in Figure 6, the standard deviation is decreased from σ_{initial} to σ_{final}. In Figure 7, it is assumed that random values are generated by normal distribution, and one seed is created around the given solution.

3.4. Mutation

In order to restrict premature convergence or stick in local optima, we apply a mutation concept. Mutation is done for each solution of pop2 through mutation

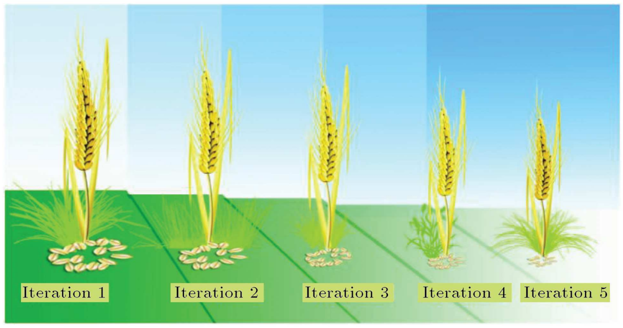

Figure 6. Schematic reduction of standard deviation.

operators. The population resulted from mutation is called pop3, and the amount of pop3 is equal to pop2, because we perform the same mutation for each solution of pop2. The mutation operators used in this study are swap, insertion and reversion. These policies are defined as below:

- Swap: The position of selected positions (i.e. Jobs 2 (position 2) and job 1(position 5) in Figure 2) are exchanged (Figure 8(b)).
- Reversion: In this policy, besides conducting swap, the positions located between the swapped positions are reversed, too (Figure 8(c)).
- Insertion: In this case, the job in second position is located immediately after the job at first location, and the other jobs are shifted to the right hand side, accordingly (see Figure 8(d)).

3.5. Affinity function

An affinity function was used for avoiding premature convergence and increasing the diversification. The affinity function allows us to generate solutions with high diversity. We consider a parameter, called a percentage of affinity, denoted by PAF, to define the percentage of good-sorted solutions remaining at each iteration. Then, the remaining capacity of the population is filled with unique solutions existing among the present solutions. If unique solutions are not enough for filling the remaining capacity of the population, we have to use repetitive solutions.

3.6. Merge

We merge pop1, pop2 and pop3, then, their fitness functions are calculated and sorted. After that, n_{pop} solutions are taken by using the sorted solutions considering the affinity function. Finally, this population is used in the next iteration as an initial population.

3.7. Termination criterion

The algorithm continues until the maximum iteration is satisfied. The framework of our proposed algorithm is illustrated in Figure 9.

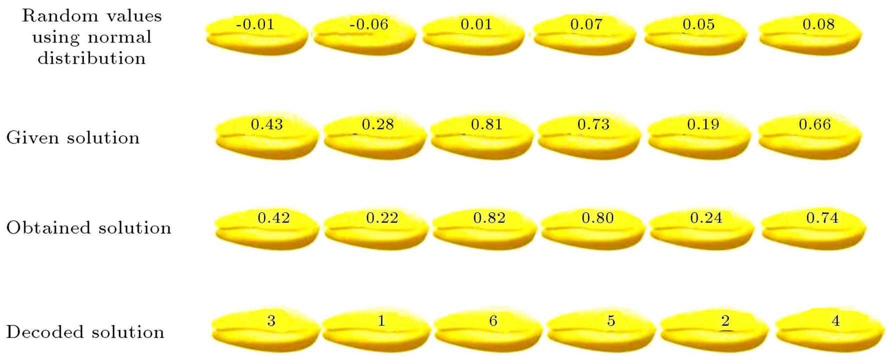

Figure 7. An example for seed reproduction.

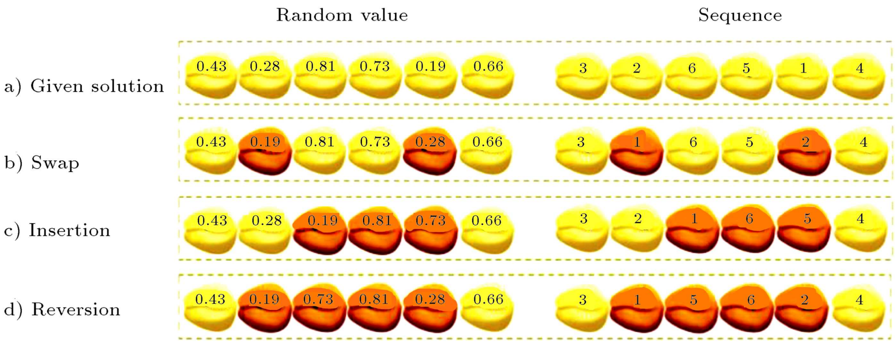

Figure 8. Swap, insertion and reversion mutations.

4. Computational experiments

The objectives of our computational experiments are two-fold. First, we want to test the efficiency of the proposed algorithm. Second, we are interested in achieving some empirical conclusions, with regard to certain characteristic factors of the mentioned flexible flow shop.

4.1. Data generation

We first describe how the instances are generated. Test instances are produced with the following combinations of job number (n) and stage (s), where $n = \{20, 40, 80, 100\}$ and $s = \{2, 4, 8\}$. We also generate groups of instances with two parallel machines per stage and groups, where the number of parallel machines at each stage is sampled from a discrete uniform distribution in the range [1,6]. The processing times and anticipatory sequence dependent setup times are basically generated from a discrete uniform distribution over the interval [1,99] and [1,50], respectively. For ready times, integers are uniformly distributed between 0 and 100. Both loading time and unloading time are generated by uniform distribution in a range between [1,15]. Travelling times between two subsequent stages are produced by uniform distribution in a range between [1,30]. The probability of rework for each operation is generated by an exponential distribution ($\lambda e^{-\lambda}$), with mean equal to 0.1 Furthermore, rework times for the jobs that need to rework the procedure are generated by a function related to the processing time of that job on a related machine (Round ($U(0.3, 0.6) \times p_{i,j}$)). The different levels of the factors result in 24 different scenarios. We produce 10 instances for each scenario. Therefore, we have 240 instances.

4.2. Parameter setting

Finding the optimum value of parameters in meta-heuristic algorithms influences the output of these algorithms. Due to being time-consuming and requiring more experiments, the majority of investigators have not applied tuning methods. Researchers usually refer to parameter values of previous studies published in this area or consider a fixed time for all of the algorithms. Before calibration of the proposed algorithm parameters, we intend to select one rule for job assignment in the conveyer. So, all algorithms with both rules, i.e. First In First Out (FIFO) and Sequence Priority (SP), are run for a random problem. Figure 10 indicates that the Sequence Priority (SP) for all algorithms statistically outperforms the other rule. Hence, for implantation of algorithms, SP is select.

Response Surface Methodology (RSM) is an optimization tool proposed by Box and Wilson [57]. The purpose of this method is to find the best value of the response. RSM is a set of statistical and mathematical techniques and useful for optimizing the stochastic

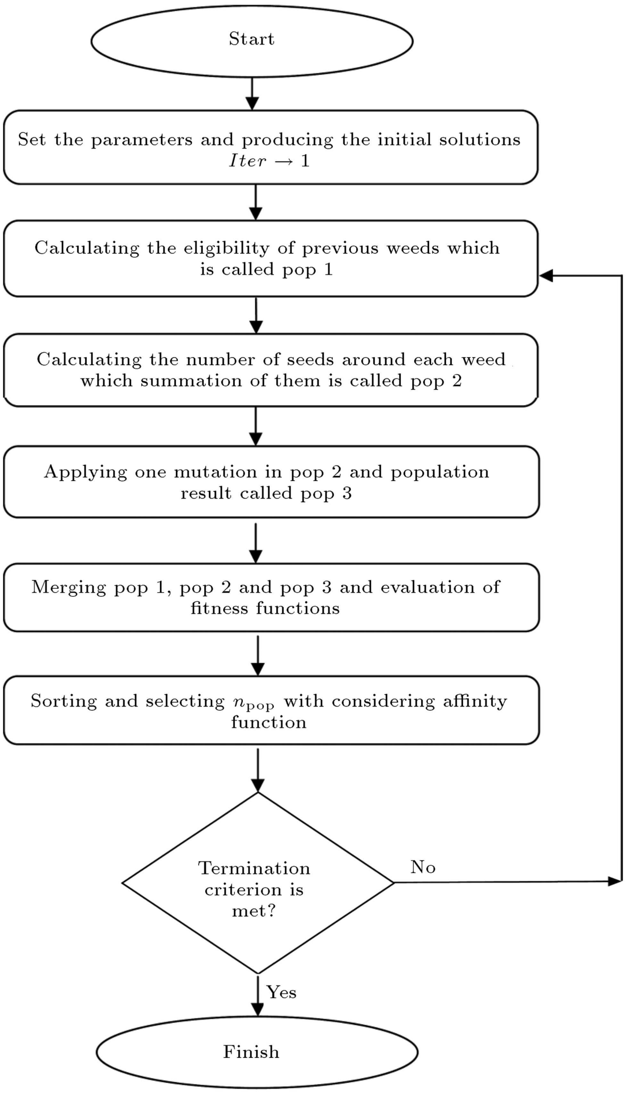

Figure 9. Schematic of EIWO.

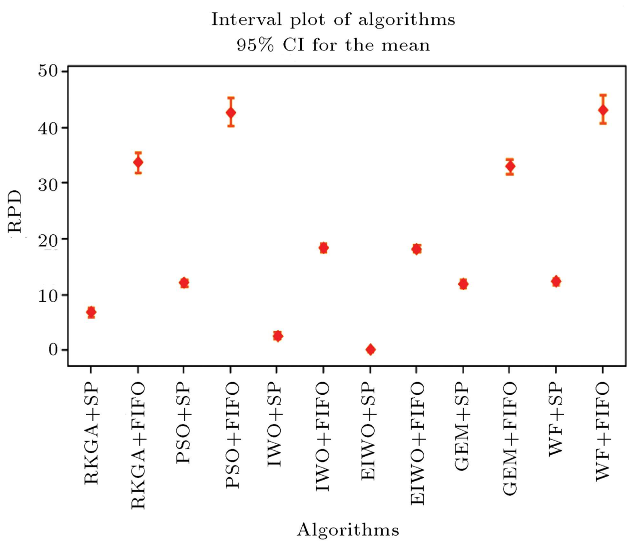

Figure 10. Comparative analogy to define the jobs how to assign to conveyors.

function, which is used in estimation of parameters in various area [58-62].

The stages of RSM are as follows: The introductory work is the selection of input variables (factors) and their levels. The second stage includes choosing an experimental design to acquire minimum variances of responses and making simulation runs considering the conditions of experimental design. The third stage is to constitute a regression meta-model and surface fitting to obtain approximate responses, and the prediction and validation of the model equation.

One of the most popular response surface designs is the Box-Behnken Design (BBD). BBD has three levels, coded -1, 0 and +1 for low, zero, and high plane, respectively [63]. The eight different factors were chosen as main variables; and determined as, $X_1, X_2, ..., X_8$, in Table 1. Independent variables used in this research are coded according to Eq. (3):

$$x_i = \frac{x_i - x_0}{\Delta x}, \tag{3}$$

where X_i and x_i indicate actual value and codified value, X_0 indicates the value of X_i at the center point, and Δx is the step change value [64]. The generalized response surface model to describe the variation of response variables is given below [65]:

$$y=\beta_0+\sum_{i=1}^{k}\beta_i X_i+\sum_{i=1}^{k}\beta_{ii}X_i^2+\sum_{i<j}\beta_{ij}X_iX_j+\varepsilon, \tag{4}$$

where y indicates the predicted response by the response surface model, β_0 is modification value, β_i ($i = 1, 2, ..., k$) indicates linear influence, β_{ij} is interaction influence and represents the quadratic influence of X_i. X_{ij} and X_j are the input variables that affect the response variable, y, which represents random error.

Each algorithm is run using different combinations of factors in Table 1, and there are three replicates for each combination. For each run, the best makespan is recorded. The response variables of experiment are then calculated with obtained makespan for each

Table 1. Coding of experimental factors and levels.

Factors	Symbol	Coded level		
		-1	0	1
n_{pop}	X_1	50	100	150
MaxIt	X_2	50	100	150
$S_{\max}$	X_3	6	8	10
$S_{\min}$	X_4	1	2	3
$\partial_{\text{initial}}$	X_5	0.2	0.25	0.3
∂_{final}	X_6	0.002	0.0015	0.001
pow	X_7	2	2.5	3
P_{AF}	X_8	0.2	0.3	0.4

Table 2. Tuned values of the parameters of the proposed algorithm.

Factor	n_{pop}	MaxIt	$S_{\max}$	$S_{\min}$	$\partial_{\text{initial}}$	∂_{final}	POW	P_{AF}
Symbol	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Optimal value	150	143	10	3	0.203	0.0014	2.01	0.20

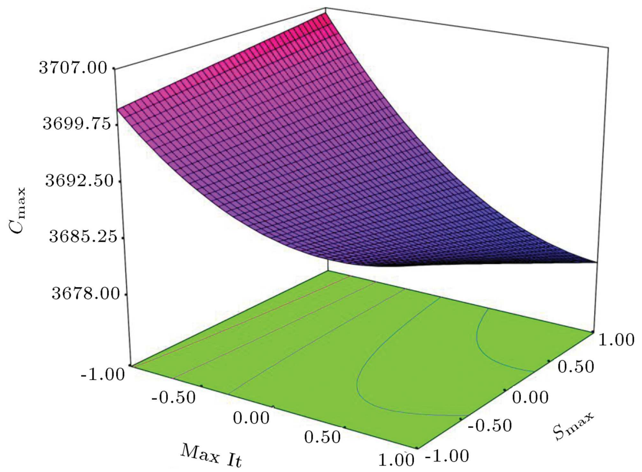

Figure 11. The interaction between MaxIt and $S_{\max}$ in EIWO.

instance. The factors which have a significant impact on the algorithm are $X_1, X_2, X_3, X_2X_3, X_2X_4, X_2X_8$, and X_2^2. For example, the interaction between X_2 and X_3 (i.e. MaxIt and $S_{\max}$) is illustrated in Figure 11.

Using significant factors, regression models are fit to the makespan value. After that, we consider the regression model as an objective function and consider the range of parameters as a constraint. Now, the mathematical model is solved and the obtained results are indicated in Table 2.

4.3. Experimental results

In this section, the Enhanced Invasive Weed Optimization (EIWO) algorithm with tuned parameters is tested with all random generated test problems. After obtaining the computational time of our proposed algorithm, the algorithms were considered as benchmarks, which were categorized in two groups. The first group contains the popular applied algorithm, as well as novel recent applied algorithms, including the Random Key Genetic Algorithm (RKGA) [40], Particle Swarm Optimization (PSO) [66-68] and Water Flow algorithms (WF) [69]. The second group is a novel metaheuristic algorithm, called the Grenade Explosion Method (GEM) [70]. Our proposed algorithm and these benchmark algorithms were run in the same time for each test problem and the results were reported. All experiments were executed on a personal computer with an Intel core2do processor of 2.53 GHz and 2 GB of RAM memory. The algorithm was coded by MATLAB 2011a language.

4.3.1. Evaluation metric

The response variable is based on a usual performance measure, which is known as RPD (relative percentage deviation) to evaluate them. The best obtained solutions for each instance are calculated. RPD is computed by the given formula as follows:

$$\text{RPD} = \frac{\text{alg}_{\text{sol}} - \text{min}_{\text{sol}}}{\text{min}_{\text{sol}}} \times 100, \tag{5}$$

where min_{sol}, the best makespan, is obtained by all the algorithms, and alg_{sol} is the makespan obtained for a given algorithm in each instance. Also, average relative percentage deviation $\overline{\text{RPD}}$is defined, according to Eq. (6):

$$\text{ARPD} = \overline{\text{RPD}} = \frac{\sum_{i=1}^{\text{No.Run}} \text{RPD}}{\text{No. Run}}. \tag{6}$$

It is clear that for each algorithm, the closer the obtained limits are to zero, with no-overlapping with other algorithms in their upper and lower limits, shows that it has yielded more proper solutions. We first analyze the makespan results and then analyze the effect of some instance factors, such as job number, stage and machine distribution.

4.3.2. Analysis of makespan

After transformation of makespan for all algorithms in each test problem, and transformation of makespan values to RPD and ARPD, the results are shown in Table 3, in terms of ARPD for a combination of job number and stage. As can be seen, the enhanced invasive weed optimization provides better results than the others. In order to verify the statistical validity of the results shown in Table 3 and to confirm the best algorithm, we have performed a Tuckey test using a 95% confidence interval. Figure 12 presents the confidence interval of algorithms at a 95% level. Considering Figure 12, it can be seen that EIWO outperforms the other algorithms statistically, and after EIWO, RKGA gets the better solution. Also, there is no significant difference between PSO and GEM.

4.3.3. Sensitive analysis

In this section, we intend to analyze the behavior of the proposed algorithm and benchmark algorithms in different situations, such as number of jobs and number of stages. We plot the Average RPDs (ARPD) of the algorithms in different levels of the number of jobs, number of stages and machines in Figures 13, 14 and 15, respectively. Towards this aim, to analyze each

Table 3. Results in terms of ARPD.

Row	Job×stage	RKGA	PSO	IWO	EIWO	GEM	WF
1	20 × 4	8.442	15.100	0.688	0.023	13.379	8.862
2	20 × 6	4.529	9.344	1.085	0.018	13.706	6.850
3	20 × 8	1.636	6.664	1.247	0.128	2.774	5.272
4	20 job	2.153	4.004	1.735	0.604	3.753	8.401
5	40 × 4	3.397	7.379	1.448	0.167	8.642	3.580
6	40 × 6	5.108	6.172	1.326	0.022	8.398	7.434
7	40 × 8	1.837	5.073	1.116	0.288	6.791	5.336
8	40 job	1.929	2.185	1.162	0.093	3.879	2.814
9	80 × 4	1.238	5.307	0.837	0.175	5.374	6.830
10	80 × 6	0.745	3.483	1.263	0.281	1.266	2.718
11	80 × 8	1.741	4.180	1.092	0.041	2.336	3.449
12	80 job	4.430	4.684	0.845	0.096	4.863	7.024
13	100 × 4	8.442	15.100	0.688	0.023	13.379	8.862
14	100 × 6	4.529	9.344	1.085	0.018	13.706	6.850
15	100 × 8	1.636	6.664	1.247	0.128	2.774	5.272
16	100 job	2.153	4.004	1.735	0.604	3.753	8.401
Total	Average	3.099	6.131	1.154	0.161	6.263	5.714

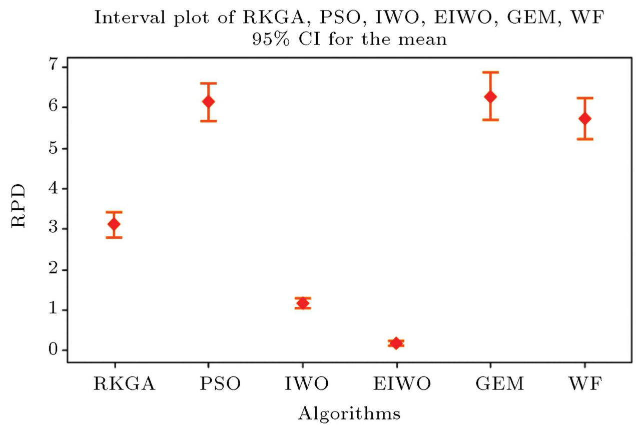

Figure 12. Means and interval plot for total problems in terms of RPD.

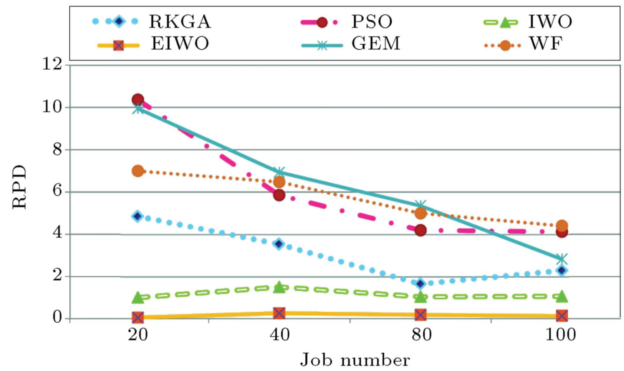

Figure 13. Means plot between the type of algorithm and number of jobs.

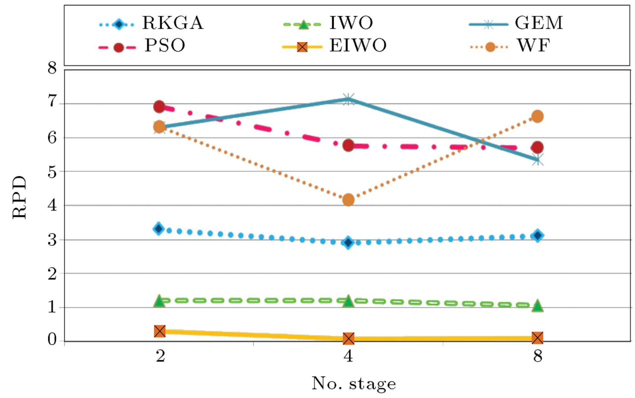

Figure 14. Means plot between the type of algorithm and number of stages.

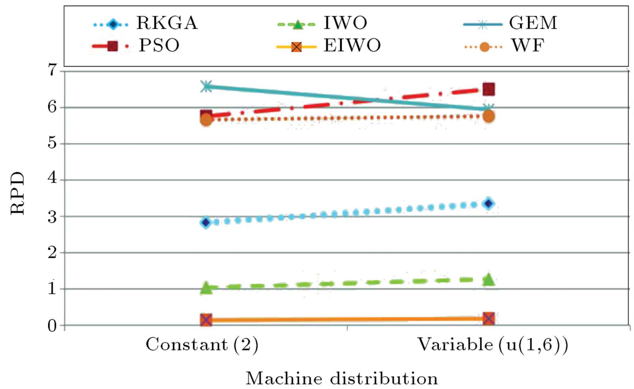

Figure 15. Means plot between the type of algorithm and machine distribution.

factor, we keep the other factors at the same value. The first point to be seen from these figures is that EIWO, in all situations, has the best performance in comparison with other algorithms. As shown in Figure 13, the results obtained from WF, PSO, and GEM algorithms have improved as the job numbers have increased. Of course, there is no significant difference between job numbers 80 and 100 for the PSO algorithm, while the trends for the other three algorithms are different. For example, the RKGB algorithm for job numbers 20 through 80, indicates a descending trend (improvement of the result). But, from job numbers 80 to 100, the trend is ascending (no improvement of the results). Moreover, there is little difference in the quality of the results for the various job numbers regarding the EIWO and IWO algorithms. In fact, a different perspective of the figure gives the idea that the analyses for different job numbers are not the same. They are rather different from each other; although, for job numbers 40 and 80, the trend of the changes remains the same. For job number 20, for example, the algorithms regarding their results are sequenced as: EIWO, IWO and RKGA followed by WF, GEM and PSO.

For job numbers 40 and 80, the first three algorithms are similar to job number 20, but the fourth ranking of the algorithm for these job numbers belongs to PSO.

The fifth ranking belongs to the WF algorithm, and the worst one for that job number goes to GEM. For job number 100, as previous job numbers, the first three algorithms are similar, but the fourth algorithm belongs to GEM, the fifth to PSO and the worst quality goes to the WF algorithm. In Figure 14, which is related to the quality of the algorithms, with respect to the number of stages at each phase, considering the RPD criterion, as the previous figure, the better algorithms (the first ones) are EIWO, IWO and RKGA, while ranking for the fourth, fifth and sixth is rather different. For stages 2, 4 and 8, the fourth, fifth, and sixth ranking belong to (PSO, WF, GEM), (GEM, PSO, WF) and (WF, PSO, GEM), respectively.

Regarding the quality of the algorithms, with respect to stages, it is seen that for the first three algorithms (RKGA, IWO, EIWO), the trend is not significant and the results for the three stages are nearly the same. Nevertheless, for the WF algorithm, the results obtained for stage 4 shows a better quality compared to stages 2 and 8. Also, for the algorithm GEM, the result for stage 8 is better than the other two stages. For the algorithm PSO, the results for stages 4 and 8 show a higher quality compared to stage 2.

Considering the distribution function of the machinery, Figure 15 reveals that the four better algorithms have almost a constant trend and the other two algorithms, in both production methods, show

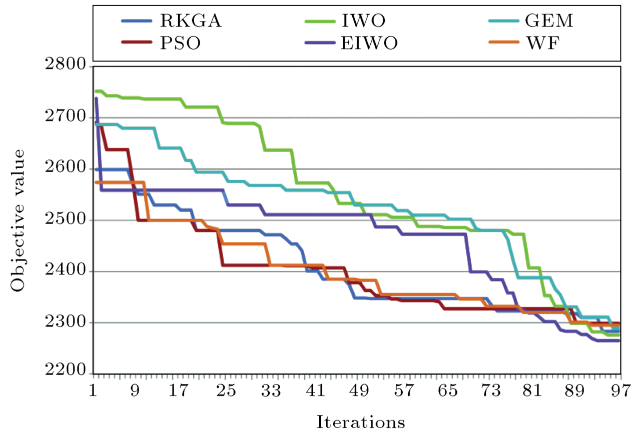

Figure 16. Algorithm convergences for an example problem (40×2).

enhancement. These algorithms are EIWO, IWO, RKGA and WF, respectively. When the number of machine are the same and equal to two, the PSO algorithm takes the fifth place in ranking, and the sixth belongs to GEM. Nevertheless, when the machinery is produced randomly, between (1 and 6) with the same distribution, the quality of the results for GEM is somewhat better than PSO. Figure 16 shows the process of algorithm convergences for a specific problem (40×2) in a specific time.

5. Conclusion and further research

To the best of our knowledge, this is the first reported investigation into solving the flexible flow shop scheduling problem considering probable rework, transportation time with a conveyor between two subsequent stages, different ready time and anticipatory sequence dependent setup times, to minimize maximum completion time. To solve the addressed problem, an effective meta-heuristic algorithm, called enhanced invasive weed optimization, is developed. Mutation operation and an affinity function are added to the original IWO for escaping local optimum and premature convergence. A comprehensive calibration by response surface methodology is done in order to achieve reliable results. The comprehensive set of computational experiments and statistical analyses for test problems under different conditions revealed that EIWO outperformed the other algorithms in terms of solution quality in the same computational time. As an interesting future research, one might work on extending this problem, considering assumptions that have attracted less interest than the others, such as random breakdown, preventive maintenance and resource dependent processing time. Another clue for future research is solving the multi-objective of the addressed problem on which we are currently working.

References

1. Adler, LB. "BPSS: A scheduling support system for the packa ng industry", *Journal of Operation Research*, **4**(1), pp. 641-648 (1993).
2. Agnetis, A., Pacifici, A., Rossi, F., Lucertini, M., Nicoletti, S. and Nicolo, F. "Scheduling of flexible flow shop in an automobile assembly plant", *European Journal of Operational Research*, **97**(2), pp. 348-62 (1997).
3. Ruiz, R. and Maroto, C. "A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility", *European Journal of Operational Research*, **169**, pp. 781-800 (2006).
4. Arthanari, T.S. and Ramamurthy, KG. "An extension of two machines sequencing problem", *Opsearch*, **8**(1), pp. 10-22 (1971).
5. Salvador, M.S. "A solution to a special case of flow shop scheduling problems", In: Elmaghraby S.E., Ed., *Symposium of the Theory of Scheduling and Its Applications.*, New York: Springer, pp. 83-91 (1973).
6. Brah, S.A. and Hunsucker, J.L. "Branch and bound algorithm for the flow shop with multiple processors", *European Journal of Operational Research*, **51**(1), pp. 88-99 (1991).
7. Moursli, O. and Pochet, Y. "A branch-and-bound algorithm for the hybrid flowshop", *International Journal of Production Economics*, **64**(1-3), pp. 113-125 (2000).
8. Gupta, J.N.D. "Two-stage, hybrid flowshop scheduling problem", *Journal of the Operational Research Society*, **39**(4), pp. 359-64 (1988).
9. Azizoglu, M., Cakmak, E. and Kondakci, S. "A flexible flowshop problem with total flow time minimization", *European Journal of Operations Research*, **132**(3), pp. 528-538 (2001).
10. Sawik, T.J. "Mixed integer programming for scheduling flexible flow lines with limited intermediate buffers", *Mathematical and Computer Modelling*, **31**(13), pp. 39-52 (2000).
11. Sawik, T.J. "Mixed integer programming for scheduling surface mount technology lines", *International Journal of Production Research*, **39**(14), pp. 3219-3235 (2001).
12. Lee, G.C. and Kim, Y.D. "A branch-and-bound algorithm for a two-stage hybrid flowshop scheduling problem minimizing total tardiness", *International Journal of Production Research*, **42**(22), pp. 4731-4743 (2004).
13. Bolat, A., Al-Harkan, I. and Al-Harbi, B. "Flowshop scheduling for three serial stations with the last two duplicate", *Computers and Operations Research*, **32**(3), pp. 647-667 (2005).
14. Sawik, T.J. "Integer programming approach to production scheduling for make-to-order manufacturing", *Mathematical and Computer Modelling*, **41**(1), pp. 99-118 (2005).
15. Haouari, M., Hidri, L. and Gharbi, A. "Optimal scheduling of a two-stage hybrid flow shop", *Mathematical Methods of Operations Research*, **64**(1), pp. 107-124 (2006).
16. Voss, S. and Witt, A. "Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-world application", *International Journal of Production Economics*, **105**(2), pp. 445-458 (2007).
17. Narasimhan, S.L. and Panwalkar, S.S. "Scheduling in a two stage manufacturing process", *International Journal of Production Research*, **22**, pp. 555-564 (1984).
18. Narasimhan, S.L. and Mangiameli, P.M. "A comparison of sequencing rules for a two-stage hybrid flow shop", *Decision Sciences*, **18**, pp. 250-265 (1987).
19. Ding, F.Y. and Kittichartphayak, D. "Heuristics for scheduling flexible flow lines", *Computers Industrial Engineering*, **26**(1), pp. 27-34 (1994).
20. Sriskandarajah, C. and Sethi, S.P. "Scheduling algorithms for flexible flowshops: Worst and average case performance", *European Journal of Operational Research*, **43**(2), pp. 143-60 (1989).
21. Braglia, M. and Petroni, A. "Data envelopment analysis for dispatching rule selection", *Production Planning and Control*, **10**(5), pp. 454-461 (1999).
22. Kadipasaoglu, S.N., Xiang, W. and Khumawala, B.M. "A comparison of sequencing rules in static and dynamic hybrid flow systems", *International Journal of Production Research*, **35**(5), pp. 1359-1384 (1997).
23. Kurz, M.E. and Askin, R.G. "Comparing scheduling rules for flexible flow lines", *International Journal of Production Economics*, **85**(3), pp. 371-388 (2003).
24. Lee, G.C. "Estimating order lead times in hybrid flowshops with different scheduling rules", *Computers and Industrial Engineering*, **56**(4), pp. 1668-1674 (2009).
25. Wang, H., Wang, D. and Yang, S. "A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problem", *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, **13**(8), pp. 763-780 (2009).
26. Qian, B., Wang, L., Huang, D.X. and Wang, X. "Multi-objective no-wait flow-shop scheduling with a memetic algorithm based on differential evolution", *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, **13**(8), pp. 847-69 (2009).
27. Goh, C., Teoh, E. and Tan, K. "A hybrid evolutionary approach for heterogeneous multiprocessor scheduling". *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, **13**(8), pp. 833-846 (2009).
28. Tavakkoli-Moghaddam, R., Khalili, M. and Naderi, B. "A hybridization of simulated annealing and

electromagnetic-like mechanism for job shop problems with machine availability and sequence-dependent setup times to minimize total weighted tardiness", *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, **13**(10), pp. 995-1006 (2009).

29. Behnamian, J., Zandieh, M. and Fatemi Ghomi, S.M.T. "Bi-objective parallel machines scheduling with sequence-dependent setup times using hybrid metaheuristics and weighted min-max technique", *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, DOI: 10.1007/s00500-010-0673-0 (2011).

30. Akhavan Niaki, S., Malaki, M. and Ershadi, M. "A particle swarm optimization approach on economic and economic-statistical designs of MEWMA control charts", *Scientia Iranica*, **18**(6), pp. 1529-1536 (2011).

31. Hajinejad, D., Salmasi, N. and Mokhtari, R. "A fast hybrid particle swarm optimization algorithm for flow shop sequence dependent group scheduling problem", *Scientia Iranica*, **18**(3), pp. 759-64 (2011).

32. Bashiri, M. and Farshbaf Geranmayeh, A. "Tuning the parameters of an artificial neural network using central composite design and genetic algorithm", *Scientia Iranica*, **18**(6), pp. 1600-1608 (2011).

33. Rabiee, M., Zandieh, M. and Jafarian, A. " Scheduling of a no-wait two-machine flow shop with sequence-dependent setup times and probable rework using robust meta-heuristics", *International Journal of Production Research*, DOI: 10.1080/00207543.2011.652747 (2012a).

34. Rabiee, M., Zandieh, M. and Ramezani, P. "Bi-objective partial flexible job shop scheduling problem: NSGA-II, NRGA, MOGA and PAES approaches", DOI: 10.1080/00207543.2011.648280 (2012b).

35. Jolai, F., Rabiee, M. and Asefi, H. " A novel hybrid meta-heuristic algorithm for a no-wait flexible flow shop scheduling problem with sequence dependent setup times", *International Journal of Production Research*, DOI: 10.1080/00207543.2011.653012 (2012).

36. Jamshidi, R., Fatemi Ghomi, S. and Karimi, B. "Multi-objective green supply chain optimization with a new hybrid memetic algorithm using the Taguchi method", *Scientia Iranica*, **19**, pp. 1876-1886 (2012).

37. Haouari, M. and M'Hallah, R. "Heuristic algorithms for the two-stage hybrid flowshop problem", *Operations Research Letters*, **21**, pp. 43-53 (1997).

38. Riane, F., Raczy, C. and Artiba, A. "Hybrid auto-adaptable simulated annealing based Heuristic", *Computers & Industrial Engineering*, **37**, pp. 277-280 (1999).

39. Negenman, E. "Local search algorithms for the multiprocessor flowshop scheduling problem", *European Journal of Operational Research*, **128**, pp. 147-158 (2001).

40. Kurz, M.E. and Askin, R.G "Scheduling flexible flow lines with sequence-dependent setup times", *European Journal of Operational Research*, **159**(1), pp. 66-82 (2004).

41. Zandieh, M., Fatemi Ghomi, S. and Moattar Husseini, S. "An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times", *Applied Mathematics and Computation*, **180**(1), pp. 111-27 (2006).

42. Naderi, B., Ruiz, R. and Zandieh, M. "Algorithms for a realistic variant of flowshop scheduling", *Computers and Operations Research*, **37**, pp. 236-246 (2010).

43. Behnamian, J. and Zandieh, M. "A discrete colonial competitive algorithm for hybrid flowshop scheduling to minimize earliness and quadratic tardiness penalties", *Expert Systems with Applications*, **38**(12), pp. 14490-14498 (2011).

44. Ribas, I., Leisten, R. and Framinan, J.M. "Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective", *Computers & Operations Research*, **37**, pp. 1439-1454 (2010).

45. Ruiz, R. and Vazquez-Rodriguez, J.A. "The hybrid flow shop scheduling problem", *European Journal of Operational Research*, **20**, pp. 1-18 (2010).

46. Naderi, B., Zandieh, M., Khaleghi Ghoshe Balagh, A. and Roshanaei, V. "An improved simulated annealing for hybrid flowshops with sequence dependent setup and transportation times to minimize total completion time and total tardiness", *Expert Systems with Applications*, **36**(6), pp. 9625-9633 (2009a).

47. Naderi, B., Zandieh, M. and Shirazi, M. "Modeling and scheduling a case of flexible flowshops: Total weighted tardiness minimization", *Computers and Industrial Engineering*, **57**, pp. 1258-1267 (2009b).

48. Guinet, A.G. and Solomon, M.M. "Scheduling hybrid flowshops to minimize maximum tardiness or maximum completion time", *International Journal of Production Research*, **34**(6), pp. 1643-1654 (1996).

49. Gupta, J.N.D., Hariri, A.M.A. and Potts, C.N. "Scheduling a two-stage hybrid flow shop with parallel machines at the first stage", *Annals of Operations Research*, **69**, pp. 171-191 (1997).

50. Shafaei, R., Rabiee, M. and Mirzaeyan, M. "An adaptive neuro fuzzy inference system for makespan estimation in multiprocessor no-wait two stage flow shop", *International Journal of Computer Integrated Manufacturing*, DOI:10.1080/0951192X.2011.597430 (2011).

51. Pinedo, M. "Scheduling: Theory, algorithms, and systems", *Prentice-Hall*, New Jersey (2002).

52. Mehrabian, A.R. and Lucas, C. "A novel numerical optimization algorithm inspired from weed colonization", *Ecological Informatics*, **1**, pp. 355-366 (2006).

53. Mehrabian, A.R. and Yousefi-Koma, A. "Optimal positioning of piezoelectric actuators on a smart fin

using bio-inspired algorithms", *Aerospace Science and Technology*, **11**(2-3), pp. 174-82 (2007).

54. Zhang, X., Wang, Y., Cui, G., Niu, Y. and Xu, J. "Application of a novel IWO to the design of encoding sequences for DNA computing", *Computers & Mathematics with Applications*, **57**(11-12),pp. 2001-2008 (2009).

55. Chowdhury, A., Ghosh, A., Giri, R. and Das, S. "Electromagnetic antenna configuration optimization using fitness adaptive differential evolution", *Swarm, Evolutionary, and Memetic Computing*, pp. 87-94 DOI: 10.1007/978-3-642-17563-3_11 (2010).

56. Ghosh, A., Das, S., Chowdhury, A. and Giri, R. "An ecologically inspired direct search method for solving optimal control problems with Bézier parameterization", *Engineering Applications of Artificial Intelligence*, doi:10.1016/j.engappai.2011.04.005 (2011).

57. Box, G.E.P. and Wilson, K.B. "On the experimental attainment of optimum conditions", *Journal of the Royal Statistical Society Series B*, **13**(1), pp. 1-38 (1951).

58. Biles, W.E. and Kleijnen, J.P.C. "A Java-based simulation manager for optimization and response surface methodology in multiple-response parallel simulation", *Winter Simulation Conference*, pp. 513-517 (1999).

59. Nicolai, R.P., Dekker, R., Piersma, N. and van Oortmarssen, G.J. "Automated response surface methodology for stochastic optimization models with unknown variance", In *Proceeding of the 2004 Winter Simulation Conference*, Piscataway, New Jersey, USA: Institute of Electrical and Electronics Engineers, Inc (2004).

60. Allen, T. and Yu, L. "Low cost response surface methods for and from simulation optimization", In *Proceeding of the 2000 Winter Simulation Conference*, 704-714. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc (2000).

61. Vahdani, B. and Zandieh, M. "Scheduling the trucks in cross docking systems: Robust meta-heuristics", *Computers & Industrial Engineering*, **58**, pp. 12-24 (2010).

62. Zandieh, M. and Karimi, N. "An adaptive multi-population genetic algorithm to solve the multi-objective group scheduling problem in hybrid flexible flow shop with sequence-dependent setup times", *Journal of Intelligent Manufacturing*, DOI: 10.1007/s10845-009-0374-7 (2009).

63. Majumder, A., Singh, A. and Goyal, A. "Application of response surface methodology for glucan production from Leuconostoc dextranicum and its structural characterization", *Carbohydrate Polym*, **75**, pp. 150-156 (2009).

64. Myers, W.R. *Encyclopedia of Biopharmaceutical Statistics*, New York: Marcel Dekker (2003).

65. Mirhosseini, H., Tan, C.P., Hamid, N.S.A., Yusof, S. and Chern, B.H. "Characterization of the influence of main emulsion components on the physicochemical properties of orange beverage emulsion using response surface methodology", *Food Hydrocolloids*, **23**, pp. 271-280 (2009).

66. Tseng, C.T. and Liao, C.J. "A particle swarm optimization algorithm for hybrid flow-shop scheduling with multiprocessor tasks", *International Journal of Production Research*, **46**(17), pp. 4655-4670 (2008).

67. Eberhart, R. and Kennedy, J. "New optimizer using particle swarm theory", In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39-43 (1995).

68. Kennedy, J. and Eberhart, R. "Particle swarm optimization", In *Proceedings of the IEEE Conference on Neural Networks*, pp. 1942-1948 (1995).

69. Pargar, F. and Zandieh, M. "Bi-criteria SDST hybrid flow shop scheduling with learning effect of setup times: water flow-like algorithm approach", **50**(10), pp. 2609-2623 (2012).

70. Ahrari, A., Shariat-Panahi, M. and Atai, A.A. "GEM: A novel evolutionary optimization method with improved neighborhood search", *Applied Mathematics and Computation*, **210**, pp. 376-386 (2009).

Biographies

Fariborz Jolai is a Professor of Industrial Engineering in the College of Engineering at the University of Tehran, Iran. He obtained his PhD degree in Industrial engineering from INPG, Grenoble, France, in 1998. His current research interests are scheduling and production planning, supply chain modeling, optimization problems under uncertainty conditions.

Reza Tavakkoli-Moghaddam is a Professor of Industrial Engineering in the College of Engineering at the University of Tehran, Iran. He obtained his PhD degree in Industrial Engineering from Swinburne University of Technology, Melbourne, Australia, in 1998, his MSc degree in Industrial Engineering from the University of Melbourne, Australia, in 1994, and his BSc degree in Industrial Engineering from Iran University of Science & Technology Tehran, Iran, in 1989. He serves on the Editorial Board of numerous reputable journals, including the International Journal of Engineering, Iranian Journal of Operations Research, Production & Operations Management (in Farsi), and Journal of Industrial Engineering (in Farsi). He is recipient of the 2009 and 2011 Distinguished Researcher Award, the 2010 Distinguished Applied Research Award at the University of Tehran, Iran, and was selected as National Iranian Distinguished Researcher for 2008 and 2010. He is also the member of the Academy of Sciences in Iran. Professor Tavakkoli-Moghaddam has published 4 books, 13 book chapters, and more than 500 papers in reputable academic journals and conferences.

Meysam Rabiee was born in Tehran, Iran, in 1985. He received his BS degree in Industrial Engineering from Bu-Ali Sina University, Hamadan, Iran, in 2008, and an M.S. degree in Industrial Engineering from K.N. Toosi University of Technology, Tehran, in 2010. He is currently Lecturer in the Department of Industrial Engineering at Bu-Ali Sina University, Tuyserkan's Engineering College, Hamadan, Iran. His research interests include scheduling, intelligent systems, metaheuristic and heuristic algorithms, supply chain and vehicle routing problems.

Elmira Gheisariha was born in Tehran, Iran, in 1984. She received her BS degree in Industrial Engineering from the College of Engineering in the University of Tehran, Iran, in 2008, and an MS degree in Industrial Engineering from the University of Tehran-Kish International Campus, Kish Island, Iran, in 2012. Her research interests include scheduling, metaheuristic and heuristic algorithms, and supply chain.