

## An order-based estimation of distribution algorithm for stochastic hybrid flow-shop scheduling problem

Sheng-yao Wang\*, Ling Wang, Min Liu and Ye Xu

*Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Automation, Tsinghua University, Beijing 10084, China*

(Received 18 January 2013; accepted 22 December 2013)

In view of the stochastic nature of the data in real-world manufacturing systems, it is crucial to develop effective algorithms to solve the scheduling problems with uncertainty. In this paper, an order-based estimation of distribution algorithm (OEDA) is proposed to solve the hybrid flow-shop scheduling problem (HFSP) with stochastic processing times. Considering the effectiveness and robustness of a schedule, it aims to minimise the makespan of the initial scenario as well as the deviation of all results of the stochastic scenarios and the initial one. To be specific, a bi-objective function is used to evaluate the individuals of the population, and a probability model is designed to describe the probability distribution of the solution space. Meanwhile, optimal computing budget allocation (OCBA) technique is employed to provide a reliable identification to the good solutions among the population. A mechanism is also presented to update the probability model with the superior individuals that are identified by the OCBA. The new individuals are generated by sampling the probability model to track the area with promising solutions. In addition, the influence of parameter setting is investigated based on Taguchi method of design-of-experiment (DOE), and a suitable parameter setting is suggested. Extensive numerical testing results and comparisons with the existing algorithm are provided, which demonstrate the effectiveness of the proposed OEDA.

**Keywords:** hybrid flow-shop scheduling problem; stochastic processing time; estimation of distribution algorithm; optimal computing budget allocation; robust scheduling

### Notation

$n$ : the number of the jobs to be processed;  
 $S$ : the number of the stages;  
 $m_k$ : the number of the machines at stage  $k$ ;  
 $J = \{J_1, J_2, \dots, J_n\}$ : the set of  $n$  jobs to be processed;  
 $R_i$ : the releasing time of job  $J_i$ ;  
 $S_{ik}$ : the starting time of job  $J_i$  at stage  $k$ ;  
 $t_{ik}$ : the processing time of job  $J_i$  at stage  $k$ ;  
 $C_{ik}$ : the completing time of job  $J_i$  at stage  $k$ ;  
 $X_{ijk}$ : a binary variable that is equal to 1 if job  $J_j$  is assigned to machine  $i$  at stage  $k$ , 0 otherwise;  
 $Y_{ijk}$ : a binary variable that is equal to 1 if job  $J_i$  precedes job  $J_j$  at stage  $k$ , 0 otherwise;  
 $L$ : a large enough constant;  
 $\alpha$ : the uncertainty degree of the processing times;  
 $I$ : the initial scenario of the problem;  
 $\xi_i(I)$ : the  $i$ th scenario with parameters sampled from  $I$ ;  
 $I_{\min}$ : the minimal scenario with the minimal processing time, i.e.,  $I_{\min} = I - \alpha I$ ;  
 $I_{\max}$ : the maximal scenario with the maximal processing time, i.e.,  $I_{\max} = I + \alpha I$ ;  
 $C_{\max}^{\zeta}(x)$ : the makespan of the solution  $x$  for the scenario  $\zeta$ ;  
 $n_{\zeta}$ : the number of the stochastic scenarios for evaluating solution  $x$ ;  
 $T_I$ : the processing time of the initial scenario;

$\lambda$ : the weight coefficient of the evaluation function;  
 $LB$ : the lower bound of the initial scenario;  
 $n_0$ : the number of initial simulation replications in the OCBA;  
 $n_x(h)$ : the number of the simulation replications for evaluating solution  $x$  in the  $h$ th iteration of the OCBA;  
 $N(h)$ : the number of already assigned simulations in the  $h$ th iteration of the OCBA;  
 $N_{\text{total}}$ : the total number of simulation replications in each generation of the OEDA;  
 $\Delta$ : the increment of the computing budget in the iteration of the OCBA;  
 $s_i$ : the standard deviation among the makespan values from the already realised simulations for solution  $i$ ;  
 $\delta_{b,i}$ : the difference in average makespan value between solution  $i$  and the best solution  $b$ ;  
 $Psize$ : the population size of the OEDA;  
 $SPsize$ : the superior sub-population size of the OEDA;  
 $P$ : the probability matrix;  
 $p_{ij}(l)$ : the element of  $P$  at generation  $l$ ;  
 $\beta$ : the learning rate for updating the probability model;  
 $\eta$ : the percentage of the superior sub-population from the whole population.

\*Corresponding author. Email: [wangshengyao@tsinghua.org.cn](mailto:wangshengyao@tsinghua.org.cn)

## 1. Introduction

The hybrid flow-shop scheduling problem (HFSP) is an extension of the classical flow-shop scheduling problem (FSP) (Wang and Zheng 2003) in the flexible manufacturing systems, which contains certain parallel machines at some processing stages. The HFSP is very close to the real manufacturing processes (Wang 2003; Desprez, Chu, and Chu 2009; Luo, Li, and Xue 2010; Wang, Xu, et al. 2012, 2013c). Many real scheduling problems in the fields of chemical, metallurgy, textile, machinery, semiconductor, logistics, architecture and papermaking can be modelled as the HFSP (Ruiz and Vázquez-Rodríguez 2010; Ribas, Leisten, and Framinan 2010). As a combination of the classical FSP and parallel machine scheduling problem, the HFSP is more difficult to solve than the classical FSP. The HFSP has been proved to be strongly non-deterministic polynomial-time (NP)-hard even in a two-stage case (Gupta 1988). Therefore, it is very important to develop effective and efficient solution algorithms for the HFSP.

Early research on the HFSP mainly focused on the exact solution methods (Moursli and Pochet 2000) and the heuristic algorithms (Santos, Hunsucker, and Deal 2001). However, the exact solution methods are inefficient in solving the large-scale problems due to the tremendous computational effort and memory requirement. Although the heuristic algorithms can obtain a solution in relatively short time, it is difficult to guarantee the quality of the solution. With the development of intelligent computing, some meta-heuristic algorithms have been proposed for the HFSP during recent years. Relevant results in this area include the following. The clonal selection principle and affinity maturation mechanism were used to design an artificial immune system algorithm (Engin and Döyen 2004). A simulated annealing-based algorithm was presented by using different dispatching rules, and a method was introduced to calculate the lower bound for the HFSP (Jin, Yang, and Ito 2006). An ant colony optimisation was presented based on local and global pheromone evaluations to solve the HFSP with the serial scheduling strategy (Alaykryan, Engin, and Döyen 2007). Besides, an improved genetic algorithm (GA) was presented by Kahraman, Engin, and Kaya (2008), and a quantum-inspired immune algorithm was proposed by hybridising the quantum rotation gate and immune algorithm (Niu, Zhou, and Ma 2009). Very recently, a particle swarm optimisation algorithm was presented by employing a bottleneck heuristic to fully exploit the bottleneck stage of the HFSP (Liao, Tjandradjaja, and Chung 2012).

In the literature, it often assumes that the processing times are deterministic. In reality, however, several types of uncertainty may affect the production situation (Chrysosouris and Subramaniam 2000, 2001). Thus, the processing times of the operations are stochastic in practice. The HFSP with stochastic processing time is named

the stochastic HFSP (SHFSP). Very recently, a GA was developed by Chaari et al. (2011) for solving the SHFSP, where a robust bi-objective evaluation function was defined to obtain robust and effective solutions. To the best of the authors' knowledge, it is the only presented algorithm for the SHFSP. The study on the SHFSP is still in its infancy. So, it is important to develop novel algorithms to solve the problem effectively.

Estimation of distribution algorithm (EDA) (Larranaga and Lozano 2002) is a novel kind of evolutionary algorithm based on statistical learning, which has gained increasing studies and wide applications during recent years. Due to different kinds of the relationships among variables, EDA has different models. Accordingly, it can be classified as univariate model, bivariate model or multivariate model. Univariate model assumes that the variables are independent of each other, e.g., the population-based incremental learning (Baluja 1994), the univariate marginal distribution algorithm (Mühlenbein and Paass 1996) and the compact GA (Harik, Lobo, and Goldberg 1999). Bivariate model assumes that each variable is associated with another one, e.g., the mutual information maximisation for input clustering (De Bonet, Isbell, and Viola 1997), the combining optimisers with mutual information trees (Baluja and Davies 1997) and the bivariate marginal distribution algorithm (Pelikan and Mühlenbein 1999). Multivariate model considers the relationship between all the variables, e.g., the factorised distribution algorithms (Mühlenbein and Mahnig 1999), the extended compact GA (Harik 1999) and the Bayesian optimisation algorithm (Pelikan 2005). So far, EDA has been applied to a variety of academic and engineering problems, such as feature selection (Saeys et al. 2003), inexact graph matching (Cesar et al. 2005), software testing (Sagarna and Lozano 2005), flow-shop scheduling (Jarboui, Eddaly, and Siarry 2009), resource-constrained project scheduling (Wang and Fang 2012), multidimensional knapsack problem (Wang, Wang, and Fang 2012), flexible job-shop scheduling (Wang, Wang, Xu, et al. 2012, 2013b; Wang, Wang, and Liu 2013) and distributed scheduling (Wang et al. 2013a).

In this paper, an order-based EDA (OEDA) is proposed to solve the SHFSP. Due to the stochastic nature of the problem, optimal computing budget allocation (OCBA) technology based on the idea of ordinal optimisation (Chen et al. 2000) is used in the algorithm. To be specific, a probability model is designed to describe the probability distribution of the solution space, and an updating mechanism is provided to update the probability model with the superior individuals. By sampling the probability model, new individuals can be generated to track the search area with promising solutions. Meanwhile, the OCBA technology is used to evaluate solutions and identify the superior ones in the random environment. In addition, the parameter

setting is investigated, and extensive numerical testing results are provided. The comparisons with the existing algorithm demonstrate the effectiveness and robustness of the proposed OEDA.

The remainder of the paper is organised as follows. In Section 2, the SHFSP is described. In Section 3, the basic EDA is introduced briefly and the OEDA for the SHFSP is presented in detail. The influence of parameter setting is investigated in Section 4, and the testing results and comparisons are provided as well. Finally, the paper ends with some conclusions and scope for future work in Section 5.

## 2. Problem description

### 2.1. The HFSP

Generally, the HFSP can be described as follows. There are a set of  $n$  jobs  $J = \{J_1, J_2, \dots, J_n\}$  to be processed at  $S$  sequential stages, where stage  $k$  ( $k = 1, 2, \dots, S$ ) contains  $m_k$  ( $m_k \geq 1$ ) identical parallel machines. Each job should be processed at all the stages sequentially, and it can be processed by any of the machines at each stage. Typically, it assumes that all the jobs are independent and available to be processed at initial time; the releasing time of any machine is not considered or set as 0; buffers between the stages are unlimited; one machine can process only one operation and one job can be processed on only one machine at a time; for all the jobs, the processing times at each stage are known in advance; the transportation times between different machines are negligible or are included in the processing times (Wang, Xu, et al. 2012, 2013c); once an operation is started, it cannot be interrupted. The HFSP is to determine both the assignment of machines at each stage and the sequence of operations on all the machines to minimise a certain scheduling objective function, e.g., the maximum complete time of all the jobs (i.e., makespan  $C_{\max}$ ). An example is illustrated in Figure 1.

The HFSP can be formulated as follows (Paternina-Arboleda et al. 2008):

$$\text{Minimise : } C_{\max} = \max C_{ik}, k = 1, 2, \dots, S; i = 1, 2, \dots, n \quad (1)$$

Subject to:

$$C_{ik} = S_{ik} + t_{ik} \quad (2)$$

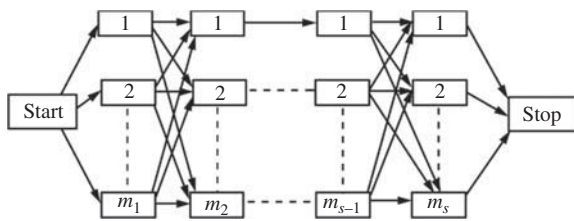


Figure 1. An example of the HFSP.

$$\sum_{i=1}^{m_k} X_{jik} = 1 \quad (3)$$

$$S_{ik} \geq C_{i,k-1}, k = 2, 3, \dots, S \quad (4)$$

$$S_{ik} \geq C_{jk} - LY_{ijk}, \text{ for all the pairs } (i, j) \quad (5)$$

$$S_{jk} \geq C_{ik} - (1 - L)Y_{ijk}, \text{ for all the pairs } (i, j) \quad (6)$$

where Equation (1) implies that the objective is to minimise the makespan, Equation (2) is for calculating the  $C_{ik}$ , Equation (3) ensures that each job is processed once at each stage, Constraints (4) and (5) ensure that one machine can process only one job at one time and Constraint (6) means that one job cannot be processed until its preceding job is finished.

### 2.2. The SHFSP

For the SHFSP, the processing times are described as random variables. Machine breakdowns and maintenance tasks are not taken into consideration in this paper. To evaluate a solution of such a stochastic problem, a scenario modelling approach is often adopted to represent the uncertain characteristic (Chaari et al. 2011). Usually, a set of stochastic scenarios are constructed, in which the processing times are sampled according to a certain probability distribution.

Same as the literature (Chaari et al. 2011), uniform distribution is considered in this paper. To be specific, the initial scenario  $I$  is a deterministic scenario, and the set of the stochastic scenarios is represented by sampling the initial scenario  $I$ .  $\xi_i(I)$  is uniformly distributed between  $[T_i - \alpha T_i, T_i + \alpha T_i]$ , where  $\alpha \in [0, 1]$  is the uncertainty degree of the processing time. For a given solution  $x$ , the average of the makespan values yielded from all the  $n_x$  scenarios is regarded as its expected makespan.

## 3. The OEDA for SHFSP

### 3.1. The basic EDA

Estimation of distribution algorithm (EDA) provides a framework for a group of probability-distribution-based algorithms. The idea is to predict and track the area with promising solutions in the search space by using a probability model (Larranaga and Lozano 2002). The general flow chart of the basic EDA is illustrated in Figure 2.

The core of the above flow chart is to estimate the probability distribution. Probability model is used in the EDA to describe the distribution of the solution space, and it is updated with the superior individuals to help the algorithm focus on the region with promising solutions. Therefore, a proper probability model and a suitable updating mechanism are crucial to the EDA, which should

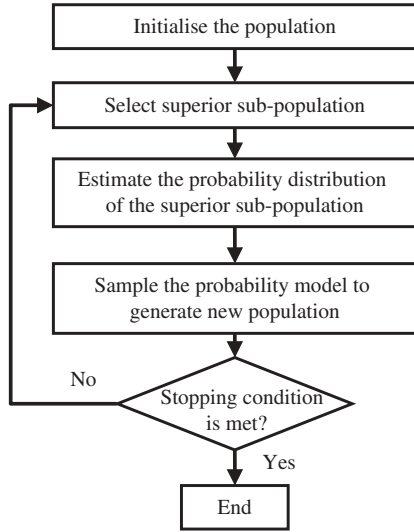


Figure 2. The general flow chart of the EDA.

be well designed for different problems. Next, the OEDA will be introduced in detail to solve the SHFSP.

### 3.2. Evaluation function

Considering the effectiveness and robustness of the schedule, an evaluation function (Chaari et al. 2011) aggregating two objectives is employed to evaluate a solution  $x$ , which is expressed as follows:

$$f(x) = \lambda \frac{C_{\max}^I(x) - LB}{LB} + (1 - \lambda) \times \frac{\sqrt{\frac{1}{n_x} \sum_{i=1}^{n_x} [C_{\max}^{\xi_i(I)}(x) - C_{\max}^I(x)]^2}}{DEV\_MAX(x^*)} \quad (7)$$

where  $\lambda \in (0, 1)$  represents the weight coefficient,  $x^*$  is the best solution obtained for the initial scenario  $I$  and  $DEV\_MAX(x^*) = \max\{(C_{\max}^I(x^*) - C_{\max}^{I_{\min}}(x^*)), (C_{\max}^{I_{\max}}(x^*) - C_{\max}^I(x^*))\}$ . The two values  $C_{\max}^{I_{\min}}(x^*)$  and  $C_{\max}^{I_{\max}}(x^*)$  are obtained by integrating the solution  $x^*$  in the minimal and maximal scenarios.

It can be seen that the schedule objective is to minimise the makespan of the initial scenario as well as the deviation of all the makespan values of the stochastic scenarios and the initial one. Clearly, both the effectiveness and robustness of a schedule are concerned. Note that  $LB$  and  $DEV\_MAX(x^*)$  are used to normalise the two different types of objectives. Obviously, the smaller the objective function value, the better is the solution.

### 3.3. Encoding and decoding

Each individual of the population is a solution of the SHFSP, expressed by an integer sequence with a length of  $n$ , which determines the processing order of jobs at the first stage. For example, a solution  $x = \{5, 2, 3, 1, 4\}$  represents that job 5 is processed first at the first stage and next are job 2, job 3 and job 1 in sequence. Job 4 is the last one to be processed at the first stage.

To decode a sequence is to assign the jobs to the machines at each stage so as to yield a feasible schedule and calculate the schedule objective value. For the SHFSP, the decoding procedure is to decide both the order of jobs and the machines assignment. For the order of jobs, the processing order of jobs at the first stage is decided referring to the order of the job numbers in the solution. From stage 2 onwards, the processing order of jobs at the current stage is decided according to their completion times at the previous stage in a non-decreasing order. As for the machines assignment, the first available machine rule (Brah and Luan 1999) is employed. That is, the job is assigned to the machine with the earliest release time.

For a given solution  $x$ , all the makespan values of the initial scenario and the stochastic scenarios are obtained with the above decoding method. Then, the objective function value  $f(x)$  of solution  $x$  is obtained.

### 3.4. Evaluation by OCBA

Based on the basic idea of order comparison and goal softening, ordinal optimisation (Ho, Sreenivas, and Vakili 1992) is an effective way to solve complex stochastic optimisation problems. Later, Chen et al. (2000, 2009) presented the OCBA technique to intelligently allocate a limited computing budget to efficiently evaluate and reliably identify good solutions. In the OEDA, a number of simulation replications,  $N_{\text{total}}$ , are given in each generation; then, OCBA is used to assign the simulation replications iteratively to each individual according to the acquired knowledge. The procedure of the OCBA is described as follows (Chen et al. 2009; Zhang, Song, and Wu 2012).

Step 1: Perform  $n_0$  simulation replications for all solutions, and calculate the average makespan value and  $s_i$  for each solution  $i$ . Set  $h = 0$ ,  $n_1(h) = n_2(h) = \dots = n_{Psize}(h) = n_0$  and  $N(h) = n_0 \cdot Psize$ .

Step 2: If  $\sum_{i=1}^{Psize} n_i(h) \geq N_{\text{total}}$ , stop the procedure and set the number of the stochastic scenarios for evaluating solution  $i$  as  $n_i = n_i(h)$ .

Step 3: Let  $N(h+1) = N(h) + \Delta$ , and compute the new budget allocation  $n_1(h+1)$ ,  $n_2(h+1)$ , ...,  $n_{Psize}(h+1)$  using the following sub-procedure.

Step 3.1: Locate the best solution and record its index as  $b$ . Then randomly find  $j$  such that  $j \neq b$ .



Step 3.2: Calculate  $n_j(h+1)$  as follows.

$$n_j(h+1) = N(h+1) \times \left\{ \sum_{i=1, i \neq b}^{Psize} \left( \frac{s_i / \delta_{b,i}}{s_j / \delta_{b,j}} \right)^2 + s_b \times \left[ \sum_{i=1, i \neq b}^{Psize} \frac{1}{s_i^2} \left( \frac{s_i / \delta_{b,i}}{s_j / \delta_{b,j}} \right)^4 \right]^{\frac{1}{2}} \right\}^{-1} \quad (8)$$

Step 3.3: Calculate  $n_i(h+1)$  (For all  $i \neq j$  and  $i \neq b$ ) as follows.

$$n_i(h+1) = n_j(h+1) \times \left( \frac{s_i / \delta_{b,i}}{s_j / \delta_{b,j}} \right)^2 \quad (9)$$

Step 3.4: Calculate  $n_b(h+1)$  as follows.

$$n_b(h+1) = n_j(h+1) \times s_b \times \left[ \sum_{i=1, i \neq b}^{Psize} \frac{1}{s_i^2} \left( \frac{s_i / \delta_{b,i}}{s_j / \delta_{b,j}} \right)^4 \right]^{\frac{1}{2}} \quad (10)$$

Step 4: Perform additional  $\max\{0, n_i(h+1) - n_i(h)\}$  simulations for evaluating solution  $i$  ( $i = 1, 2, \dots, Psize$ ) and update the average makespan value and  $s_i$  for each solution  $i$  if necessary.

Step 5: Let  $n_i(h+1) = \max\{n_i(h+1), n_i(h)\}$ . Let  $h = h+1$  and go to Step 2.

For a solution  $x$  in the population, its simulation replications  $n_x$  is obtained by the OCBA. Then, its objective value  $f(x)$  is calculated as in Section 3.2. From Equation (9), it can be seen that very few replications are provided by OCBA for the inferior solution with a high  $\delta_{b,i}$ . Thus, the inferior solution will have few chances to be evaluated in the subsequent allocation. In contrast, more simulation replications are allocated by OCBA to the solutions with high  $s_i$  to promote the evaluation accuracy of performance.

To balance the simulation budget and the accuracy, both  $n_0$  and  $\Delta$  should not be too small or too large. According to the suggested range of  $n_0$  and  $\Delta$  (Chen et al. 2000, 2009), it sets  $n_0 = 10$  and  $\Delta = 10$  in this paper.

### 3.5. Probability model and updating mechanism

The EDA produces a new population by sampling the probability model. In this paper, the probability model is designed as a probability matrix  $P$ , where the element  $p_{ij}(l)$  represents the probability that job  $J_j$  appears before or in position  $i$  of the solution sequence at generation  $l$ . The value of  $p_{ij}$  implies the importance of a job when deciding the jobs order. For all  $i$  and  $j$ ,  $p_{ij}$  is initialised to  $p_{ij}(0) = 1/n$ , which ensures that the whole solution space can be sampled uniformly.

In each generation of the OEDA, the new individuals are generated via sampling the probability matrix  $P$ . For every position  $i$ , job  $J_j$  is selected with the probability  $p_{ij}$ . If job  $J_j$  has already appeared, the whole  $j$ th column of probability matrix  $P$  will be set as zero and all the elements of  $P$  will be normalised to maintain that each row sums up to 1. An individual is constructed until all the jobs appear in the sequence. In such a way,  $Psize$  individuals are generated.

Next, it determines the superior sub-population with the best  $SPsize$  solutions identified by OCBA. Then, the probability matrix  $P$  is updated as follows:

$$p_{ij}(l+1) = (1 - \beta)p_{ij}(l) + \frac{\beta}{i \times SPsize} \sum_{s=1}^{SPsize} I_{ij}^s, \forall i, j \quad (11)$$

where  $\beta \in (0, 1)$  is the learning rate of  $P$ , and  $I_{ij}^s$  is the following indicator function of the  $s$ th individual in the superior sub-population.

$$I_{ij}^s = \begin{cases} 1, & \text{if job } J_j \text{ appears before or in position } i \\ 0, & \text{else} \end{cases} \quad (12)$$

The above updating process can be regarded as a kind of increased learning, where the second term on the right hand side of Equation (11) represents the learning information from the superior sub-population.

### 3.6. Procedure of the OEDA for SHFSP

With the above design, the procedure of the OEDA for solving the SHFSP is illustrated in Figure 3.

In each generation,  $Psize$  individuals are generated by sampling the probability matrix after the matrix is initialised. Then, the best  $SPsize$  individuals identified by OCBA are selected to update the probability model. If the stopping condition is met, it will stop the algorithm and output the optimal solution; otherwise, it will generate  $Psize$  new individuals for the next generation. Since the probability model is updated with elite individuals, it helps the algorithm track the search region with promising solutions. So, it is expected to solve the SHFSP effectively.

### 3.7. Computational complexity analysis

For each generation of the OEDA, its computational complexity can be roughly analysed as follows.

For the sampling process, every position is generated with the roulette strategy by sampling the probability matrix  $P$ . A sequence is constructed with the complexity  $O(n^2)$  and

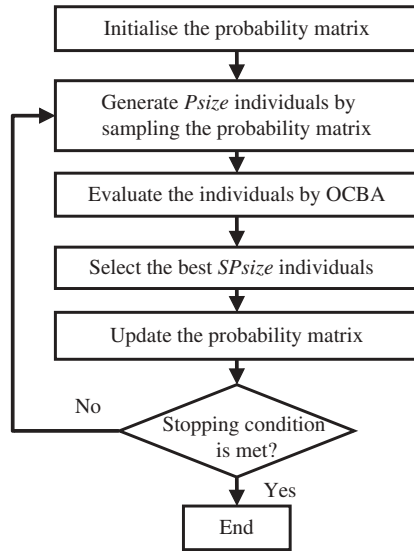


Figure 3. The procedure of the OEDA for the SHFSP.

$Psize$  individuals can be generated with the complexity  $O(n^2 Psize)$ .

For the updating process, first it is with the computational complexity  $O(Psize \log Psize)$  by using the quick sorting method to select the best  $SPsize$  individuals from population; then, it is with the complexity  $O[n(SPsize + n)]$  to update all the  $n \times n$  elements of  $P$ . Thus, the computational complexity for updating process is  $O[n(SPsize + n) + Psize \log Psize]$ .

It can be seen that the complexity of the proposed OEDA is not large and the algorithm could be efficient.

#### 4. Numerical results and comparisons

To test the performance of the proposed OEDA, numerical tests are carried out by using the benchmarks provided by Chaari et al. (2011). Table 1 lists the configurations and the  $LB$  values of these benchmarks. The algorithm is coded in C and is run on Thinkpad T420 with a 2.3GHz processor/2GB RAM. For each benchmark instance, it sets

Table 1. Configurations and  $LB$  values of the benchmarks.

No	Benchmark	$n$	$S$	$LB$ value
1	j10c5a2	10	5	88
2	j10c5b1	10	5	130
3	j10c5c1	10	5	68
4	j10c5d1	10	5	66
5	j15c5a1	15	5	178
6	j15c5b1	15	5	170
7	j15c5c1	15	5	85
8	j15c5d1	15	5	167
9	2center20job	20	2	161
10	2center50job	50	2	256
11	2center100job	100	2	402

the total number of evaluations as  $10^5$ , which is used as the stopping condition for the OEDA.

##### 4.1. Parameters setting

The OEDA contains four key parameters:  $Psize$  (population size),  $\eta$  (percentage of superior sub-population from population,  $\eta = SPsize/Psize \times 100$ ),  $N_{total}$  (the total computing budget in each generation) and  $\beta$  (learning rate of  $P$ ). To investigate the influence of these parameters on the performance of the OEDA, the Taguchi method of design-of-experiment (DOE) (Montgomery 2005) is implemented by using a moderate-scale instance No. 9 with  $\alpha = 10\%$  and  $\lambda = 0.5$ . Combinations of different values of these parameters are listed in Table 2.

For each combination, the OEDA is run 10 times independently. The average response variable (ARV) value is the average objective function value of 10 runs, i.e.,  $ARV = \sum_{i=1}^{10} f_i(x) / 10$ . Clearly, the smaller the ARV, the better is the combination. Considering 4 factor levels for each parameter, an orthogonal array  $L_{16}(4^4)$  is shown in Table 3 together with the obtained ARV values.

Table 2. Combinations of parameter values.

Parameter	Factor level			
	1	2	3	4
$Psize$	30	40	50	60
$\eta$	10	20	30	40
$N_{total}$	750	1000	1250	1500
$\beta$	0.1	0.2	0.3	0.4

Table 3. Orthogonal array and ARV values.

No. of experiment	Factor				ARV( $\times 10^{-3}$ )
	$Psize$	$\eta$	$N_{total}$	$\beta$	
1	1	1	1	1	95.23
2	1	2	2	2	95.08
3	1	3	3	3	91.44
4	1	4	4	4	97.83
5	2	1	2	3	87.16
6	2	2	1	4	88.09
7	2	3	4	1	101.83
8	2	4	3	2	104.15
9	3	1	3	4	103.52
10	3	2	4	3	102.33
11	3	3	1	2	81.54
12	3	4	2	1	76.17
13	4	1	4	2	85.98
14	4	2	3	1	89.73
15	4	3	2	4	99.75
16	4	4	1	3	106.56

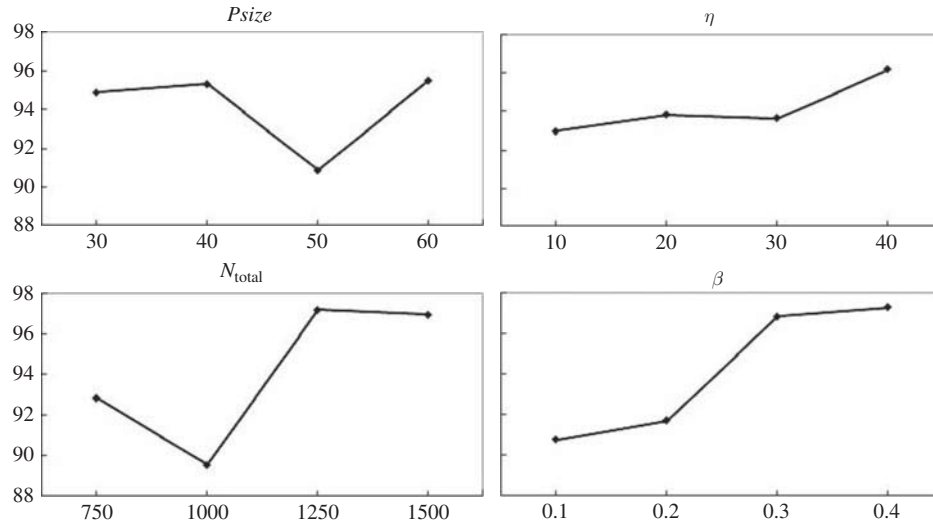


Figure 4. Factor level trend of the OEDA.

According to the orthogonal table, the trend of each factor level is illustrated in Figure 4. Then, the response value of each parameter is figured out to analyse the significance rank of each parameter. Besides, the best level of each parameter is obtained, e.g., level 3 of  $Psize$ , level 1 of  $\eta$ , level 2 of  $N_{total}$  and level 1 of  $\beta$ . The results are listed in Table 4.

From Table 4, it can be seen that  $N_{total}$  is the most significant one among the four parameters. That is, the total computing budget in each generation is crucial to the OEDA. A small value of  $N_{total}$  could lead to insufficient evaluation for the solutions of the population. With a fixed total number of evaluations, however, a larger value of  $N_{total}$  will lead to insufficient evolution with fewer generations. The significance of the learning rate  $\beta$  of  $P$  ranks the second. A large value of  $\beta$  could lead to premature convergence. Besides, the population size  $Psize$  ranks the third. An appropriate population size ensures that the whole solution space can be sampled sufficiently. As for  $\eta$ , it has the slightest influence, while a smaller value is preferable to build the probability model. According to the above analysis, the parameter

setting for the OEDA is suggested as  $Psize = 50$ ,  $\eta = 10$ ,  $N_{total} = 1000$  and  $\beta = 0.1$ , which will be used for the following tests.

#### 4.2. Testing results and comparisons

To compare the proposed OEDA to the GA (Chaari et al. 2011), the best solution obtained by the algorithm is evaluated extra 100 times to confirm the accuracy as given in the literature (Chaari et al. 2011). Furthermore, the number of evaluations in the GA (Chaari et al. 2011) is no less than  $2 \times 10^6$  and it is only  $10^5$  in the OEDA. Besides, to study the influence of the OCBA on the OEDA, the results of the algorithm without the OCBA are also given. For a fair comparison, the algorithm without the OCBA has the same parameter setting as the OEDA, and 20 stochastic scenarios are evaluated for each solution in the population.

The detailed results are summarised in Tables 5–7 for the benchmarks with the uncertainty degree 10%, 25% and 50%, respectively. The column ' $C_{max}^I$ ' presents the best makespan value of the initial scenario. The column ' $STD$ ' presents the deviation of all the makespan values of the stochastic scenarios and the initial one, which is calculated as follows:

$$STD = \sqrt{\frac{1}{100} \sum_{i=1}^{100} (C_{max}^{\xi_i(I)} - C_{max}^I)^2} \quad (13)$$

The column ' $AVG$ ' presents the average makespan value of 100 stochastic scenarios. The column ' $DEV$ ' (in %) presents the deviation between  $AVG$  and  $C_{max}^I$ , which is calculated as follows:

Table 4. Response value ( $\times 10^{-3}$ ) and rank of each parameter.

Level	Parameter			
	$Psize$	$\eta$	$N_{total}$	$\beta$
1	94.8950	<b>92.9725</b>	92.8550	<b>90.7400</b>
2	95.3075	93.8075	<b>89.5400</b>	91.6875
3	<b>90.8900</b>	93.6400	97.2100	96.8725
4	95.5050	96.1775	96.9925	97.2975
Delta	4.6150	3.2050	7.6700	6.5575
Rank	3	4	1	2

Note: The bold values mean the best results.

Table 5. Experimental results for  $\alpha = 10\%$ .

No.	$\lambda$	GA				EDA without OCBA				OEDA			
		$C_{\max}^I$	STD	AVG	DEV	$C_{\max}^I$	STD	AVG	DEV	$C_{\max}^I$	STD	AVG	DEV
1	1	<b>88</b>	8.83	96.04	9.13	<b>88</b>	4.94	89.08	1.23	<b>88</b>	4.42	87.04	-1.09
	0.5	93	4.24	96.29	3.53	89	2.28	87.96	-1.17	<b>88</b>	2.17	87.21	-0.90
	0	124	3.73	124.83	0.66	105	2.25	104.14	-0.82	<b>101</b>	1.88	100.16	-0.83
2	1	<b>130</b>	7.06	136.66	5.12	<b>130</b>	3.05	128.00	-1.53	<b>130</b>	3.01	127.80	-1.69
	0.5	134	4.34	137.42	2.55	131	2.75	129.50	-1.34	<b>130</b>	2.46	128.49	-1.16
	0	146	4.01	148.1	1.43	135	2.55	134.07	-0.69	<b>130</b>	2.05	129.45	-0.42
3	1	<b>68</b>	7.80	75.37	10.83	<b>68</b>	2.14	67.03	-1.43	<b>68</b>	2.01	67.02	-1.44
	0.5	76	3.60	78.64	3.47	72	2.09	71.40	-0.83	<b>71</b>	1.99	70.76	-0.34
	0	89	2.80	89.39	0.43	82	1.89	81.74	-0.31	<b>80</b>	1.83	80.02	0.03
4	1	<b>66</b>	7.92	73.53	11.40	<b>66</b>	2.07	65.17	-1.26	<b>66</b>	2.06	64.94	-1.61
	0.5	72	2.70	73.01	1.40	69	1.95	69.59	0.85	<b>68</b>	1.64	68.38	0.56
	0	79	1.91	79.74	0.93	74	1.72	73.50	-0.68	<b>70</b>	1.63	69.76	-0.34
5	1	<b>178</b>	9.30	186.48	4.76	<b>178</b>	5.08	180.94	1.65	<b>178</b>	3.86	175.56	-1.37
	0.5	185	5.13	188.6	1.94	183	3.33	181.36	-0.90	<b>180</b>	3.26	178.38	-0.90
	0	205	5.07	205.2	0.09	189	3.16	187.68	-0.70	<b>186</b>	2.65	184.42	-0.85
6	1	<b>170</b>	9.86	179.31	5.47	<b>170</b>	3.23	168.00	-1.18	<b>170</b>	3.17	167.97	-1.19
	0.5	175	6.82	181.05	3.45	171	3.15	169.21	-1.05	<b>170</b>	2.83	168.13	-1.10
	0	188	4.79	190.47	1.31	176	2.58	174.84	-0.66	<b>175</b>	2.52	174.03	-0.55
7	1	<b>85</b>	10.82	95.47	12.31	<b>85</b>	3.19	85.95	1.12	<b>85</b>	2.80	85.76	0.89
	0.5	94	3.85	95.92	2.04	92	2.24	90.81	-1.29	<b>91</b>	2.52	90.30	-0.77
	0	114	2.83	113.61	-0.34	100	2.14	100.47	0.47	<b>98</b>	2.00	97.72	-0.29
8	1	<b>167</b>	8.63	175.08	4.83	<b>167</b>	3.00	165.51	-0.89	<b>167</b>	2.88	165.47	-0.92
	0.5	174	6.04	179.1	2.93	170	2.72	168.87	-0.66	<b>167</b>	2.66	165.93	-0.64
	0	183	4.71	184.22	0.66	175	2.68	173.88	-0.64	<b>170</b>	2.59	169.10	-0.53
9	1	<b>161</b>	8.97	169.23	5.11	<b>161</b>	3.02	159.87	-0.70	<b>161</b>	2.93	159.84	-0.72
	0.5	165	4.04	168.3	2.00	163	3.44	161.34	-1.02	<b>161</b>	2.35	160.18	-0.51
	0	177	3.93	175.23	-1.00	168	3.02	166.77	-0.73	<b>166</b>	1.99	165.59	-0.25
10	1	<b>256</b>	10.98	266.75	4.19	<b>256</b>	2.55	257.25	0.49	<b>256</b>	2.07	256.40	0.16
	0.5	269	4.66	272.97	1.47	263	2.50	262.58	-0.16	<b>261</b>	2.43	261.13	0.05
	0	298	4.27	298.29	0.09	273	2.40	273.22	0.08	<b>272</b>	2.28	272.00	0
11	1	<b>402</b>	15.15	416.92	3.71	<b>402</b>	2.84	403.35	0.34	<b>402</b>	2.76	402.47	0.12
	0.5	417	5.86	421.82	1.15	412	2.74	411.72	-0.07	<b>410</b>	2.66	410.40	0.10
	0	433	4.96	435.74	0.63	417	2.70	416.32	-0.16	<b>414</b>	2.46	413.72	-0.07

Note: The bold values mean the best results.



Table 6. Experimental results for  $\alpha = 25\%$ .

No	$\lambda$	GA				EDA without OCBA				OEDA			
		$C_{\max}^I$	STD	AVG	DEV	$C_{\max}^I$	STD	AVG	DEV	$C_{\max}^I$	STD	AVG	DEV
1	1	<b>88</b>	9.63	96.19	9.30	<b>88</b>	7.06	89.6	1.82	<b>88</b>	<b>6.02</b>	<b>89.53</b>	1.74
	0.5	93	5.36	96.07	3.30	90	5.07	89.27	-0.81	<b>88</b>	<b>4.08</b>	<b>87.51</b>	-0.56
2	0	109	4.18	107.85	-1.05	91	4.09	90.82	-0.20	<b>90</b>	<b>3.93</b>	<b>89.89</b>	-0.12
	1	<b>130</b>	7.10	135.4	4.15	<b>130</b>	5.68	128.46	-1.18	<b>130</b>	<b>4.97</b>	<b>128.08</b>	-1.48
3	0.5	134	5.51	136.21	1.64	<b>130</b>	4.93	129.18	-0.63	<b>130</b>	<b>4.86</b>	<b>128.77</b>	-0.95
	0	139	5.43	140.89	1.35	135	5.13	135.19	0.14	<b>130</b>	<b>4.64</b>	<b>130.13</b>	0.10
4	1	<b>68</b>	9.88	77.14	13.44	<b>68</b>	3.80	70.31	3.40	<b>68</b>	<b>3.63</b>	<b>69.75</b>	2.57
	0.5	75	4.25	77.56	3.41	72	3.66	71.68	-0.44	<b>70</b>	<b>3.25</b>	<b>70.38</b>	0.54
5	0	84	3.87	83.60	-0.47	78	3.45	78.33	0.42	<b>77</b>	<b>3.16</b>	<b>77.15</b>	0.19
	1	<b>66</b>	8.07	73.39	11.19	<b>66</b>	3.80	68.56	3.88	<b>66</b>	<b>3.11</b>	<b>67.73</b>	2.62
6	0.5	72	3.42	72.21	0.29	69	3.07	69.73	1.06	<b>68</b>	<b>2.64</b>	<b>68.29</b>	0.43
	0	78	2.83	76.81	-1.52	73	2.59	73.51	0.70	<b>71</b>	<b>2.25</b>	<b>70.65</b>	-0.49
7	1	<b>178</b>	12.78	188.32	5.79	<b>178</b>	8.79	181.87	2.17	<b>178</b>	<b>8.20</b>	<b>181.08</b>	1.73
	0.5	183	9.28	188.88	3.21	181	6.49	179.87	-0.62	<b>180</b>	<b>6.23</b>	<b>179.09</b>	-0.51
8	0	208	8.42	209.74	0.80	185	6.34	183.18	-0.98	<b>184</b>	<b>6.06</b>	<b>182.5</b>	-0.82
	1	<b>170</b>	9.36	177.41	4.35	<b>170</b>	6.05	168.49	-0.89	<b>170</b>	<b>5.90</b>	<b>167.86</b>	-1.26
9	0.5	171	7.68	176.05	2.95	<b>170</b>	5.84	168.8	-0.71	<b>170</b>	<b>5.66</b>	<b>168.73</b>	-0.75
	0	182	6.96	185.06	1.68	174	5.67	173.31	-0.40	<b>170</b>	<b>5.31</b>	<b>169.77</b>	-0.14
10	1	<b>85</b>	11.95	96.33	13.32	<b>85</b>	4.33	87.93	3.44	<b>85</b>	<b>4.29</b>	<b>87.13</b>	2.51
	0.5	93	3.68	95.62	2.81	89	3.12	90.24	1.39	<b>88</b>	<b>3.10</b>	<b>88.88</b>	1.00
11	0	102	3.48	102.26	0.25	95	3.03	94.46	-0.57	<b>94</b>	<b>2.72</b>	<b>94.14</b>	0.15
	1	<b>167</b>	9.31	174.25	4.34	<b>167</b>	6.79	168.37	0.82	<b>167</b>	<b>6.07</b>	<b>165.79</b>	-0.72
12	0.5	172	6.57	175.07	1.78	<b>167</b>	5.76	166.55	-0.27	<b>167</b>	<b>5.71</b>	<b>165.84</b>	-0.69
	0	180	5.98	180.58	0.32	174	5.76	173.91	-0.05	<b>170</b>	<b>5.6</b>	<b>169.07</b>	-0.55
13	1	<b>161</b>	11.00	170.02	5.60	<b>161</b>	6.17	162.52	0.94	<b>161</b>	<b>5.09</b>	<b>161.94</b>	0.58
	0.5	164	6.27	167.72	2.26	163	5.53	163.89	0.55	<b>162</b>	<b>4.96</b>	<b>162.36</b>	0.22
14	0	178	6.18	178.76	0.42	168	5.42	167.09	-0.54	<b>165</b>	<b>4.76</b>	<b>164.60</b>	-0.24
	1	<b>256</b>	13.50	268.60	4.92	<b>256</b>	6.82	259.65	1.43	<b>256</b>	<b>5.57</b>	<b>258.46</b>	0.96
15	0.5	265	6.89	269.14	1.56	264	5.29	265.57	0.59	<b>263</b>	<b>5.21</b>	<b>264.45</b>	0.55
	0	284	5.90	286.04	0.71	274	5.21	276.00	0.73	<b>271</b>	<b>5.08</b>	<b>269.56</b>	-0.53
16	1	<b>402</b>	16.50	417.64	3.89	<b>402</b>	7.70	405.89	0.97	<b>402</b>	<b>6.53</b>	<b>404.81</b>	0.70
	0.5	415	8.64	421.4	1.54	413	6.33	411.56	-0.35	<b>412</b>	<b>6.02</b>	<b>411.47</b>	-0.13
17	0	441	7.68	442.02	0.23	419	5.98	418.70	-0.07	<b>416</b>	<b>5.51</b>	<b>416.41</b>	0.10

Note: The bold values mean the best results.

Table 7. Experimental results for  $\alpha = 50\%$ .

No	$\lambda$	GA				EDA without OCB				OEDA			
		$C'_{\max}$	STD	AVG	DEV	$C'_{\max}$	STD	AVG	DEV	$C'_{\max}$	STD	AVG	DEV
1	1	<b>88</b>	13.12	98.9	12.38	<b>88</b>	9.51	89.66	1.89	<b>88</b>	<b>8.24</b>	<b>89.05</b>	1.19
	0.5	93	8.34	97.52	4.86	<b>88</b>	8.30	88.89	1.01	<b>88</b>	<b>7.69</b>	<b>87.24</b>	-0.86
	0	111	8.15	109.95	-0.94	93	7.91	92.02	-1.05	<b>91</b>	<b>7.32</b>	<b>90.87</b>	-0.14
2	1	<b>130</b>	11.50	136.12	4.70	<b>130</b>	10.56	133.14	2.42	<b>130</b>	<b>9.78</b>	<b>128.74</b>	-0.97
	0.5	<b>130</b>	10.03	135.17	3.97	<b>130</b>	9.53	131.31	1.01	<b>130</b>	<b>8.99</b>	<b>131.00</b>	0.77
	0	137	9.85	138.81	1.32	135	9.36	134.18	-0.60	<b>134</b>	<b>8.95</b>	<b>134.27</b>	0.20
3	1	<b>68</b>	13.12	79.76	17.29	<b>68</b>	8.89	74.40	9.41	<b>68</b>	<b>8.16</b>	<b>74.01</b>	8.84
	0.5	73	7.28	77.61	6.31	72	6.54	74.81	3.90	<b>71</b>	<b>5.71</b>	<b>72.97</b>	2.77
	0	79	6.14	80.66	2.10	76	5.77	76.75	0.97	<b>75</b>	<b>5.67</b>	<b>73.90</b>	-1.46
4	1	<b>66</b>	10.06	74.37	12.68	<b>66</b>	7.65	71.77	8.74	<b>66</b>	<b>6.37</b>	<b>70.68</b>	7.09
	0.5	72	5.28	73.84	2.55	70	5.06	71.52	2.17	<b>69</b>	<b>4.75</b>	<b>70.26</b>	1.83
	0	79	5.02	78.31	-0.87	73	4.94	73.80	1.10	<b>72</b>	<b>4.45</b>	<b>72.39</b>	0.54
5	1	<b>178</b>	16.52	186.86	4.97	<b>178</b>	14.70	181.74	2.10	<b>178</b>	<b>12.64</b>	<b>180.09</b>	1.17
	0.5	<b>178</b>	14.76	184.8	3.82	<b>178</b>	12.11	180.83	1.59	<b>178</b>	<b>11.62</b>	<b>176.70</b>	-0.73
	0	186	13.46	187.97	1.05	186	12.01	184.04	-1.05	<b>185</b>	<b>11.50</b>	<b>184.74</b>	-0.14
6	1	<b>170</b>	13.18	177.51	4.41	<b>170</b>	11.68	171.94	1.14	<b>170</b>	<b>11.15</b>	<b>168.50</b>	-0.88
	0.5	<b>170</b>	12.24	175.9	3.47	<b>170</b>	11.01	170.73	0.43	<b>170</b>	<b>10.31</b>	<b>168.94</b>	-0.62
	0	182	11.71	183.71	0.93	174	10.92	173.29	-0.41	<b>171</b>	<b>9.28</b>	<b>171.75</b>	0.44
7	1	<b>85</b>	13.78	97.68	14.91	<b>85</b>	9.35	92.02	8.26	<b>85</b>	<b>7.91</b>	<b>90.81</b>	6.84
	0.5	91	7.29	96.35	5.87	90	6.84	94.43	4.92	<b>88</b>	<b>6.43</b>	<b>91.13</b>	3.56
	0	100	6.76	99.93	-0.07	97	5.64	95.22	-1.84	<b>94</b>	<b>5.58</b>	<b>94.00</b>	0
8	1	<b>167</b>	13.36	172.15	3.08	<b>167</b>	13.16	169.26	1.35	<b>167</b>	<b>10.91</b>	<b>165.24</b>	-1.05
	0.5	<b>167</b>	11.81	169.83	1.69	<b>167</b>	10.89	169.12	1.27	<b>167</b>	<b>9.33</b>	<b>165.44</b>	-0.93
	0	173	11.43	174.12	0.64	170	10.36	168.57	-0.84	<b>167</b>	<b>9.33</b>	<b>165.70</b>	-0.78
9	1	<b>161</b>	12.59	168.98	4.95	<b>161</b>	10.40	165.16	2.58	<b>161</b>	<b>9.91</b>	<b>164.97</b>	2.47
	0.5	162	10.51	167.61	3.46	162	10.35	164.39	1.48	<b>161</b>	<b>9.45</b>	<b>161.92</b>	0.57
	0	169	9.37	169.52	0.30	167	9.04	167.37	0.22	<b>166</b>	<b>8.52</b>	<b>165.09</b>	-0.55
10	1	<b>256</b>	17.44	270.99	5.85	<b>256</b>	12.27	263.79	3.04	<b>256</b>	<b>11.62</b>	<b>263.21</b>	2.82
	0.5	264	10.47	270.21	2.35	263	10.43	265.64	1.00	<b>262</b>	<b>10.35</b>	<b>265.5</b>	1.34
	0	273	10.07	277.49	1.64	272	9.43	274.36	0.87	<b>270</b>	<b>9.19</b>	<b>271.67</b>	0.62
11	1	<b>402</b>	19.92	418.71	4.15	<b>402</b>	14.11	409.97	1.98	<b>402</b>	<b>13.70</b>	<b>409.65</b>	1.90
	0.5	405	15.98	414.83	2.42	405	13.17	412.11	1.76	<b>404</b>	<b>12.94</b>	<b>409.88</b>	1.46
	0	432	11.98	434.99	0.69	425	11.44	426.95	0.46	<b>413</b>	<b>10.84</b>	<b>414.69</b>	0.41

Note: The bold values mean the best results.

$$DEV = \frac{AVG - C_{\max}^I}{C_{\max}^I} \times 100 \quad (14)$$

According to Tables 5–7, as the value of  $\lambda$  decreases, the values of  $C_{\max}^I$  increase and the values of  $STD$  decrease. The reason is that it only minimises the  $C_{\max}^I$  when  $\lambda = 1$ , while it only minimises the  $STD$  when  $\lambda = 0$ . For  $\lambda = 0.5$ , it makes a trade-off between the two objectives.

From Tables 5–7, it can also be seen that the OEDA performs better than both the GA and the EDA without OCBA for solving the SHFSP. The  $C_{\max}^I$  values and the  $AVG$  values by the OEDA are smallest among the three algorithms for all the instances, showing that the OEDA is most effective in solving the initial scenario and the stochastic scenarios. In addition, the  $STD$  values by the OEDA are also the smallest for all the instances, showing that the OEDA is most robust on the data uncertainty. To make easy reading of Tables 5–7, the relative percentage

improvement of the OEDA to the GA is shown in Table 8, where  $PI_{C_{\max}^I}$ ,  $PI_{STD}$  and  $PI_{AVG}$  are calculated as follows:

$$PI_{C_{\max}^I} = \frac{C_{\max}^I(GA) - C_{\max}^I(EDA)}{C_{\max}^I(GA)} \times 100 \quad (15)$$

$$PI_{STD} = \frac{STD(GA) - STD(EDA)}{STD(GA)} \times 100 \quad (16)$$

$$PI_{AVG} = \frac{AVG(GA) - AVG(EDA)}{AVG(GA)} \times 100 \quad (17)$$

From Table 8, it can be seen that the values of  $PI_{C_{\max}^I}$ ,  $PI_{STD}$  and  $PI_{AVG}$  are all non-negative, which means that the OEDA can improve all the results of the GA for all the instances.

Table 8. Relative percentage improvement of the OEDA to the GA.

No	$\lambda$	$\alpha = 10\%$			$\alpha = 25\%$			$\alpha = 50\%$		
		$PI_{C_{\max}^I}$	$PI_{STD}$	$PI_{AVG}$	$PI_{C_{\max}^I}$	$PI_{STD}$	$PI_{AVG}$	$PI_{C_{\max}^I}$	$PI_{STD}$	$PI_{AVG}$
1	1	0	49.94	9.37	0	37.49	6.92	0	37.20	9.96
	0.5	5.38	48.82	9.43	5.38	23.88	8.91	5.38	7.79	10.54
	0	18.55	49.60	19.76	17.43	5.98	16.65	18.02	10.18	17.35
2	1	0	57.37	6.48	0	30	5.41	0	14.96	5.42
	0.5	2.99	43.32	6.50	2.99	11.80	5.46	0	10.37	3.09
	0	10.96	48.88	12.59	6.47	14.55	7.64	2.19	9.14	3.27
3	1	0	74.23	11.08	0	63.26	9.58	0	37.80	7.21
	0.5	6.58	44.72	10.02	6.67	23.53	9.26	2.74	21.57	5.98
	0	10.11	34.64	10.48	8.33	18.35	7.72	5.06	7.65	8.38
4	1	0	73.99	11.68	0	61.46	7.71	0	36.68	4.96
	0.5	5.56	39.26	6.34	5.56	22.81	5.43	4.17	10.04	4.85
	0	11.39	14.66	12.52	8.97	20.49	8.02	8.86	11.35	7.56
5	1	0	58.49	5.86	0	35.84	3.84	0	23.49	3.62
	0.5	2.70	36.45	5.42	1.64	32.87	5.18	0	21.27	4.38
	0	9.27	47.73	10.13	11.54	28.03	12.99	0.54	14.56	1.72
6	1	0	67.85	6.32	0	36.97	5.38	0	15.40	5.08
	0.5	2.86	58.50	7.14	0.58	26.30	4.16	0	15.77	3.96
	0	6.91	47.39	8.63	6.59	23.71	8.26	6.04	20.75	6.51
7	1	0	74.12	10.17	0	64.10	9.55	0	42.60	7.03
	0.5	3.19	34.55	5.86	5.38	15.76	7.05	3.30	11.80	5.42
	0	14.04	29.33	13.99	7.84	21.84	7.94	6.00	17.46	5.93
8	1	0	66.63	5.49	0	34.80	4.86	0	18.34	4.01
	0.5	4.02	55.96	7.35	2.91	13.09	5.27	0	13.89	2.58
	0	7.10	45.01	8.21	5.56	6.35	6.37	3.47	18.37	4.84
9	1	0	67.34	5.55	0	53.73	4.75	0	21.29	2.37
	0.5	2.42	41.83	4.82	1.22	20.89	3.20	0.62	10.09	3.39
	0	6.21	49.36	5.50	7.30	22.98	7.92	1.78	9.07	2.61
10	1	0	81.15	3.88	0	58.74	3.78	0	33.37	2.87
	0.5	2.97	47.85	4.34	0.75	24.38	1.74	0.76	1.15	1.74
	0	8.72	46.60	8.81	4.58	13.90	5.76	1.10	8.74	2.10
11	1	0	81.78	3.47	0	60.42	3.07	0	31.22	2.16
	0.5	1.68	54.61	2.71	0.72	30.32	2.36	0.25	19.02	1.19
	0	4.39	50.40	5.05	5.67	28.26	5.79	4.40	9.52	4.67
Mean		4.49	52.19	8.03	3.76	29.91	6.60	2.26	17.94	5.05

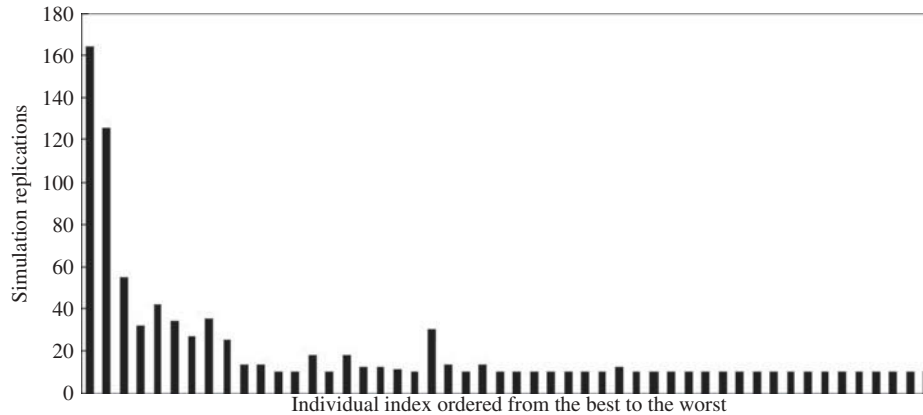


Figure 5. Computing budget allocation of OCBA in the OEDA.

Besides, the allocation of computing budget in a certain generation of the OEDA is shown in Figure 5, where the solutions among the population are ordered from the best to the worst. It implies that the OCBA is able to intelligently allocate computing budget for solutions. Assigning more for superior solutions, it helps the OEDA to effectively update the probability model and produce more promising solutions.

All in all, the OEDA is an effective and robust algorithm for solving the SHFSP. Its superiority owes to the following aspects. (1) With the well-designed probability model and the suitable updating mechanism, it is helpful to explore the search space effectively. (2) With the OCBA, it is helpful to identify superior solutions reasonably and update the probability model effectively. (3) With the suitable parameter setting, it is helpful to perform order-based EDA search.

## 5. Conclusions

This was the first research work to propose a novel EDA-based algorithm by employing the OCBA technique to solve the HFSP with stochastic processing time effectively. A bi-objective evaluation function was employed considering the effectiveness and robustness of the schedule. A specific probability model was designed, and a mechanism was presented for the OEDA to solve the problem. The OCBA was employed to evaluate solutions and identify superior ones for updating the model. The influence of parameter setting was investigated, and the suitable values were recommended. Extensive numerical testing results verified the contribution of employing the OCBA in the OEDA. Moreover, the comparison with the existing genetic algorithm demonstrated the effectiveness and robustness of the proposed OEDA. The OEDA enriches the tool-kit of SHFSP and can be generalised to other kinds of problems. The future work is to consider the SHFSP with other kinds of distribution laws or other schedule objective functions.

The scheduling problems with machine breakdowns or maintenance tasks as well as the problems with fuzzy or interval processing times are also worthy of being studied.

## Acknowledgements

The authors would like to thank Sondes Chaabane for providing the benchmark problems.

## Funding

This research is partially supported by the National Key Basic Research and Development Program of China [grant number 2013CB329503], the National Science Foundation of China [grant number 61174189] and the Doctoral Program Foundation of Institutions of Higher Education of China [grant number 20130002110057].

## References

- Alaykiran, K., O. Engin, and A. Döyen. 2007. "Using Ant Colony Optimization to Solve Hybrid Flow Shop Scheduling Problems." *International Journal of Advanced Manufacturing Technology* 35 (5–6): 541–550.
- Baluja, S. 1994. *Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*. Technical Report CMU-CS-94-163. Pittsburgh, PA: Carnegie Mellon University.
- Baluja, S., and S. Davies. 1997. "Using Optimal Dependency – Trees for Combinatorial Optimization: Learning the Structure of the Search Space." In *Proceedings of the 14th International Conference on Machine Learning*, No. CMU-CS-97-107, San Francisco, CA, 30–38. Pittsburgh, PA: Department of Computer Science, Carnegie Mellon University.
- Brah, S. A., and L. L. Luan. 1999. "Heuristics for Scheduling in a Flowshop with Multiple Processors." *European Journal of Operational Research* 113 (1): 113–122.
- Cesar, R. M., E. Bengoetxea, I. Bloch, and P. Larranaga. 2005. "Inexact Graph Matching for Model-Based Recognition: Evaluation and Comparison of Optimization Algorithms." *Pattern Recognition* 38 (11): 2099–2113.

- Chaari, T., S. Chaabane, T. Loukil, and D. Trentesaux. 2011. "A Genetic Algorithm for Robust Hybrid Flow Shop Scheduling." *International Journal of Computer Integrated Manufacturing* 24 (9): 821–833.
- Chen, C. H., J. Lin, E. Yücesan, and S. E. Chick. 2000. "Simulation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization." *Discrete Event Dynamic Systems* 10 (3): 251–270.
- Chen, C. H., E. Yücesan, L. Dai, and H. C. Chen. 2009. "Optimal Budget Allocation for Discrete-Event Simulation Experiments." *IEEE Transactions* 42 (1): 60–70.
- Chrysosolouris, G., and V. Subramaniam. 2000. "Dynamic Scheduling of Manufacturing Job Shops Using Extreme Value Theory." *Production Planning & Control* 11 (2): 122–132.
- Chrysosolouris, G., and V. Subramaniam. 2001. "Dynamic Scheduling of Manufacturing Job Shops Using Genetic Algorithms." *Journal of Intelligent Manufacturing* 12 (3): 281–293.
- De Bonet, J. S., C. L. Isbell Jr, and P. Viola. 1997. "MIMIC: Finding Optima by Estimating Probability Densities." In *Advances in Neural Information Processing Systems*, edited by M. C. Mozer, M. I. Jordan, and T. Petsche, 424–430. Cambridge: MIT Press.
- Desprez, C., F. Chu, and C. Chu. 2009. "Minimising the Weighted Number of Tardy Jobs in a Hybrid Flow Shop with Genetic Algorithm." *International Journal of Computer Integrated Manufacturing* 22 (8): 745–757.
- Engin, O., and A. Döyen. 2004. "A New Approach to Solve Hybrid Flow Shop Scheduling Problems by Artificial Immune System." *Future Generation Computer Systems* 20 (6): 1083–1095.
- Gupta, J. N. D. 1988. "Two-Stage Hybrid Flow Shop Scheduling Problem." *Journal of the Operational Research Society* 39 (4): 359–364.
- Harik, G. 1999. *Linkage Learning via Probabilistic Modeling in the ECGA*. IlliGAL Report No. 99010. Urbana: Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.
- Harik, G. R., F. G. Lobo, and D. E. Goldberg. 1999. "The Compact Genetic Algorithm." *IEEE Transactions on Evolutionary Computation* 3 (4): 287–297.
- Ho, Y. C., R. Sreenivas, and P. Vakili. 1992. "Ordinal Optimization of Discrete Event Dynamic Systems." *Discrete Event Dynamic Systems* 2 (2): 61–88.
- Jarbou, B., M. Eddaly, and P. Siarry. 2009. "An Estimation of Distribution Algorithm for Minimizing the Total Flowtime in Permutation Flowshop Scheduling Problems." *Computer & Operations Research* 36 (9): 2638–2646.
- Jin, H., Z. Yang, and Z. Ito. 2006. "Metaheuristic Algorithms for the Multistage Hybrid Flowshop Scheduling Problem." *International Journal of Production Economics* 100 (2): 322–334.
- Kahraman, C., O. Engin, and I. Kaya. 2008. "An Application of Effective Genetic Algorithms for Solving Hybrid Flow Shop Scheduling Problems." *International Journal of Computational Intelligence Systems* 1 (2): 134–147.
- Larranaga, P., and J. A. Lozano. 2002. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Boston, MA: Kluwer Press.
- Liao, J. C., E. Tjandradjaja, and T. P. Chung. 2012. "An Approach Using Particle Swarm Optimization and Bottleneck Heuristic to Solve Hybrid Flow Shop Scheduling Problem." *Applied Soft Computing* 12 (6): 1755–1764.
- Luo, X. G., W. Li, and D. Y. Xue. 2010. "Optimal Resource Allocation for Hybrid Flow Shop in One-of-a-Kind Production." *International Journal of Computer Integrated Manufacturing* 23 (2): 146–154.
- Montgomery, D. C. 2005. *Design and Analysis of Experiments*. Phoenix, AZ: John Wiley and Sons.
- Moursli, O., and Y. Pochet. 2000. "A Branch-and-Bound Algorithm for the Hybrid Flowshop." *International Journal of Production Economics* 64 (1–3): 113–125.
- Mühlenbein, H., and T. Mahnig. 1999. "Convergence Theory and Applications of the Factorized Distribution Algorithm." *Journal of Computing and Information Technology* 7 (1): 19–32.
- Mühlenbein, H., and G. Paass. 1996. "From Recombination of Genes to the Estimation of Distributions I: Binary Parameters." *Lecture Notes in Computer Science* 1141: 178–187.
- Niu, Q., T. Zhou, and S. Ma. 2009. "A Quantum-Inspired Immune Algorithm for Hybrid Flow Shop with Makespan Criterion." *Journal of Universal Computer Science* 15 (4): 765–785.
- Paternina-Arboleda, C. D., J. R. Montoya-Torres, M. J. Acero-Dominguez, and M. C. Herrera-Hernandez. 2008. "Scheduling Jobs on a k-Stage Flexible Flow-Shop." *Annals of Operations Research* 164 (1): 29–40.
- Pelikan, M. 2005. *Hierarchical Bayesian Optimization Algorithm*. Berlin: Springer.
- Pelikan, M., and H. Mühlenbein. 1999. "The Bivariate Marginal Distribution Algorithm." In *Advances in Soft Computing: Engineering Design and Manufacturing*, edited by J. M. Benítez, 521–535. London: Springer-Verlag.
- Ribas, I., R. Leisten, and J. M. Framinan. 2010. "Review and Classification of Hybrid Flow Shop Scheduling Problems from a Production System and a Solutions Procedure Perspective." *Computers & Operations Research* 37 (8): 1439–1454.
- Ruiz, R., and J. A. Vázquez-Rodríguez. 2010. "The Hybrid Flow Shop Scheduling Problem." *European Journal of Operational Research* 205 (1): 1–18.
- Saeys, Y., S. Degroove, D. Aeyels, Y. Van de Peer, and P. Rouze. 2003. "Fast Feature Selection Using a Simple Estimation of Distribution Algorithm: A Case Study on Splice Site Prediction." *Bioinformatics* 19 (suppl. 2): 179–188.
- Sagarna, R., and J. Lozano. 2005. "On the Performance of Estimation of Distribution Algorithms Applied to Software Testing." *Applied Artificial Intelligence* 19: 457–489.
- Santos, D. L., J. L. Hunsucker, and D. E. Deal. 2001. "On Makespan Improvement in Flow Shops with Multiple Processors." *Production Planning & Control* 12 (3): 283–295.
- Wang, L. 2003. *Shop Scheduling with Genetic Algorithms*. Beijing: Tsinghua University & Springer Press.
- Wang, L., and C. Fang. 2012. "An Effective Estimation of Distribution Algorithm for the Multi-Mode Resource-Constrained Project Scheduling Problem." *Computers & Operations Research* 39 (2): 449–460.
- Wang, L., S. Y. Wang, and C. Fang. 2012. "An Effective Hybrid EDA-Based Algorithm for Solving Multidimensional Knapsack Problem." *Expert Systems with Applications* 39 (5): 5593–5599.
- Wang, L., S. Y. Wang, and M. Liu. 2013. "A Pareto-Based Estimation of Distribution Algorithm for the Multi-Objective Flexible Job-Shop Scheduling Problem." *International Journal of Production Research* 51 (12): 3574–3592.
- Wang, L., S. Y. Wang, Y. Xu, G. Zhou, and M. Liu. 2012. "A Bi-Population Based Estimation of Distribution Algorithm for



- the Flexible Job-Shop Scheduling Problem.” *Computers & Industrial Engineering* 62 (4): 917–926.
- Wang, L., Y. Xu, G. Zhou, S. Y. Wang, and M. Liu. 2012. “A Novel Decoding Method for the Hybrid Flow-Shop Scheduling Problem with Multiprocessor Tasks.” *International Journal of Advanced Manufacturing Technology* 59 (9–12): 1113–1125.
- Wang, L., and D. Z. Zheng. 2003. “An Effective Hybrid Heuristic for Flow Shop Scheduling.” *International Journal of Advanced Manufacturing Technology* 21 (1): 38–44.
- Wang, S. Y., L. Wang, M. Liu, and Y. Xu. 2013a. “An Effective Estimation of Distribution Algorithm for Solving the Distributed Permutation Flow-Shop Scheduling Problem.” *International Journal of Production Economics* 145 (1): 387–396.
- Wang, S. Y., L. Wang, M. Liu, and Y. Xu. 2013b. “An Effective Estimation of Distribution Algorithm for the Flexible Job-Shop Scheduling Problem with Fuzzy Processing Time.” *International Journal of Production Research* 51 (12): 3778–3793.
- Wang, S. Y., L. Wang, M. Liu, and Y. Xu. 2013c. “An Enhanced Estimation of Distribution Algorithm for Solving Hybrid Flow-Shop Scheduling Problem with Identical Parallel Machines.” *International Journal of Advanced Manufacturing Technology*. doi:[10.1007/s00170-013-4819-y](https://doi.org/10.1007/s00170-013-4819-y).
- Zhang, R., S. J. Song, and C. Wu. 2012. “A Two-Stage Hybrid Particle Swarm Optimization Algorithm for the Stochastic Job Shop Scheduling Problem.” *Knowledge-Based Systems* 27: 393–406.