Journal of the Operational Research Society (2015), 1–15 0160-5682/15

www.palgrave-journals.com/jors/

A fuzzy logic-based hybrid estimation of distribution algorithm for distributed permutation flowshop scheduling problems under machine breakdown

Kai Wang[1], Yun Huang[2*] and Hu Qin[3]

[1]*Economics and Management School, Wuhan University, Wuhan, China;* [2]*School of Business, Macau University of Science and Technology, Macau, China; and* [3]*School of Management, Huazhong University of Science and Technology, Wuhan, China*

As the research interest in distributed scheduling is growing, distributed permutation flowshop scheduling problems (DPFSPs) have recently attracted an increasing attention. This paper presents a fuzzy logic-based hybrid estimation of distribution algorithm (FL-HEDA) to address DPFSPs under machine breakdown with makespan criterion. In order to explore more promising search space, FL-HEDA hybridises the probabilistic model of estimation of distribution algorithm with crossover and mutation operators of genetic algorithm to produce new offspring. In the FL-HEDA, a novel fuzzy logic-based adaptive evolution strategy (FL-AES) is adopted to preserve the population diversity by dynamically adjusting the ratio of offspring generated by the probabilistic model. Moreover, a discrete-event simulator that models the production process under machine breakdown is applied to evaluate expected makespan of offspring individuals. The simulation results show the effectiveness of FL-HEDA in solving DPFSPs under machine breakdown.

Journal of the Operational Research Society advance online publication, 22 July 2015; doi:10.1057/jors.2015.50

Keywords: distributed permutation flowshop; estimation of distribution algorithm; fuzzy logic; machine breakdown; makespan

1. Introduction

To enhance cost competitiveness and market responsiveness under the trend of globalisation, traditional centralised manufacturing systems have been evolving into distributed ones during the last decades (Leitão, 2009; De Giovanni and Pezzella, 2010; Naderi and Azab, 2014). Therefore, distributed manufacturing that adopts multi-factory networks to provide the manufacturers with more flexibility has recently attracted an increasing attention (Chan and Chung, 2013; Chan *et al*, 2013). In a distributed manufacturing system, scheduling is generally a two-step decision-making process that involves both job allocation to suitable factories and job scheduling in a factory. Thus, scheduling in such a multi-factory environment is obviously more complicated than scheduling in a single-factory environment.

Because of a variety of applications in manufacturing systems, the permutation flowshop scheduling problem (PFSP) has received considerable attention in recent years (Jarboui *et al*, 2009; Luo *et al*, 2012). In the permutation flowshop, jobs are released to a single factory and processed in the same order of machines. The PFSP has been known to be strongly NP-complete for more than two machines (Garey *et al*, 1976). As a generalisation of the classical PFSP, the distributed permutation flow shop scheduling problem (DPFSP) has recently been proposed by Naderi and Ruiz (2010). Unlike the PFSP that is assumed to process all jobs at the same factory, the DPFSP allows allocation of jobs to a set of distributed factories. Since the DPFSP can be reduced to the regular PFSP if only one factory is considered, it is considerably more difficult to solve than the PFSP and accordingly has been proven NP-complete (Naderi and Ruiz, 2010; Lin *et al*, 2013).

Although the DPFSP has both academic and industrial significance, relatively few studies have been devoted to developing effective approaches to this problem. Because of great difficulty in solving the DPFSP, heuristics and meta-heuristics are commonly applied to obtain near-optimal solutions within reasonable computation time. Gao and Chen (2011) integrated genetic algorithm (GA) with a local search scheme to minimise the makespan of DPFSP. More recently, to address the same scheduling problem, Gao *et al* (2013) developed a new tabu search heuristic, in which a novel strategy that exchanged sub-sequences of two factories was applied to generate neighbour solutions. Lin *et al* (2013) presented a modified iterated greedy algorithm to address the DPFSP. This algorithm adopted a new acceptance criterion that was

*Correspondence: Yun Huang, School of Business, Macau University of Science and Technology, Taipa, Macau 999078, China.
E-mail: yuhuang@must.edu.mo

commonly applied in the simulated annealing algorithm to diversify the search area. Wang *et al* (2013c) studied the DPFSP with makespan criterion and developed an efficient estimation of distribution algorithm to solve both small-sized and large-sized DPFSP instances. Hatami *et al* (2013) proposed three constructive algorithms and a variable neighbourhood descent algorithm to deal with a special case of DPFSP, namely the distributed assembly PFSP. This problem had more than one production factories to manufacture product components and a single assembly factory to produce the final products.

Despite an increasing effort devoted to the DPFSP, the existing research works commonly assume static multi-factory environments, in which no uncertainties occur during the schedule execution phase. Real-world manufacturing, however, is dynamic and stochastic in nature and suffers a variety of uncertainties (Aytug *et al*, 2005; Ouelhadj and Petrovic, 2009), including rush orders, machine breakdown, uncertain processing times, and so on. These uncertainties may inevitably result in deviations from the planned schedules and eventually affect the manufacturing system performance (Wang and Choi, 2014). A search of available literature shows that no research efforts have been devoted to addressing DPFSPs with consideration of uncertainties.

This study aims to develop an effective scheduling metaheuristic that can generate high-quality solutions to DPFSPs in dynamic manufacturing environments. The estimation of distribution algorithm (EDA), proposed by Mühlenbein and Paass (1996), is a relatively new evolutionary algorithm (EA). Different from conventional EAs, it generates the offspring by a probabilistic model which is learned from selected promising individuals of the previous population (Wang *et al*, 2014). Since EDA is capable of inheriting good genes from parents, it has been successfully applied to a variety of complex optimisation problems in both academic and industrial areas during recent years, including PFSP (Chen and Chen, 2013). To minimise the total flow time of PFSPs, Jarboui *et al* (2009) and Zhang and Li (2011) applied EDA to find better solutions. Ceberio *et al* (2014) developed a novel generalised Mallows EDA, which adopted the distance-based exponential probabilistic model to explore promising solution regions. Very recently, Shen *et al* (2015) proposed an effective bi-population EDA for no-idle PFSPs with total tardiness minimisation. In this algorithm, different probability models for global exploration and local exploitation were established to generate two sub-populations. In addition, to enhance EDA performance for addressing the PFSPs, hybridisations of EDA, and other EAs have been investigated. Liu *et al* (2011) incorporated an EDA operator into particle swarm optimisation (PSO) to learn from the collective experience of swarm. Tzeng *et al* (2012) integrated EDA with ant colony optimisation to obtain good solutions to PFSPs.

In this paper, we present an effective fuzzy logic-based hybrid EDA (FL-HEDA) to minimise the makespan of DPFSP under machine breakdown. To the best of our knowledge, no related studies on DPFSPs under uncertainties have been found in the available literature. The uniqueness of FL-HEDA lies in a fuzzy logic-based adaptive evolution strategy (FL-AES) and an embedded simulation model. To avoid premature stagnation, FL-HEDA employs a novel FL-AES to integrate the probabilistic model of EDA with crossover and mutation of GA for population generation. The proposed FL-AES is the first attempt in the literature to apply fuzzy logic to dynamically adjust the ratio of offspring produced by EDA. Moreover, to evaluate the expected makespan of offspring individuals, a discrete-event simulator is incorporated into the EDA to mimic the production process under machine breakdown.

The remainder of this paper is organised as follows. Section 2 focuses on the mathematical formulation of DPFSP. Section 3 describes the framework of the proposed FL-HEDA and its details are explained in Section 4. To validate the effectiveness of FL-HEDA under machine breakdown, FL-HEDA is compared with several scheduling algorithms and Section 5 presents the experimental results. Section 6 concludes the paper and suggests possible directions for future work.

2. Problem description

DPFSP is a relatively new scheduling problem that allows to schedule a set $J = \{1, 2, \ldots, n\}$ of n jobs in f distributed identical factories. As shown in Figure 1, each of the factories consists of the same set $M = \{M_1, M_2, \ldots, M_m\}$ of m machines arranged in series, and is therefore capable of processing all jobs. A job j, $j \in J$, contains m operations and is successively processed in the machine order of $M_1, M_2, \ldots, M_m$ in any one of these factories. In a real-world distributed permutation flowshop, however, scheduling is a reactive process in which unexpected events may render the generated schedules infeasible. Since machine breakdown commonly occurs after long-time operation in the manufacturing industry (Chen, 2006; Wang and Choi, 2012), this paper considers the DPFSP under machine breakdown.

To model machine breakdown in a real-world manufacturing environment, three parameters, namely mean time to repair (*MTTR*), mean time between failure (*MTBF*), and machine breakdown level (*BL*), may be adopted (Adibi *et al*, 2010; Wang *et al* (2013a); Zhang *et al*, 2013). *MTTR* denotes the average of repair time to restart a failed machine, while *MTBF* indicates the average of processing time between two consecutive failures on a machine. *BL*, defined as $BL = MTTR/(MTTR + MTBF)$, describes the failure frequency of a machine; it equals

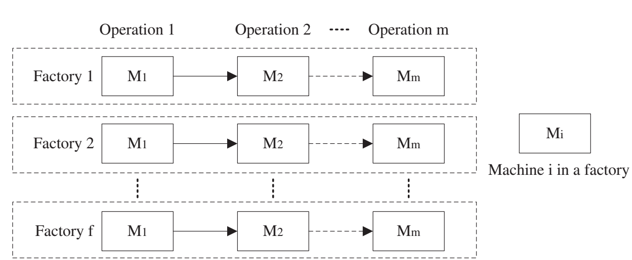

Figure 1 The distributed permutation flowshop environment.

0 when no machine breakdown occurs, and increases with the percentage of unavailable machine operating time. In this study, we simulate machine breakdown using two parameters, namely *BL* and *MTTR*. For example, *MTBF* is computed as $MTBF = MTTR/BL - MTTR = 380$ time units when $BL = 0.05$ and $MTTR = 20$ time units. Therefore, job processing on a machine would be continued over an average of 380 time units before machine breakdown occurs, and each breakdown requires an average of 20 time units for repair.

To solve the DPFSP under machine breakdown, the following assumptions are considered:

Assumption 1 *All the jobs arrive at the distributed permutation flowshop simultaneously;*

Assumption 2 *All the machines in a factory are immediately available when jobs are arrived;*

Assumption 3 *Factory switching is not allowed. Once a job is allocated to a certain factory, its operations can only be finished* in the *same factory;*

Assumption 4 *Preemption is not permitted. The operation cannot be interrupted by any other operations once it has been processed on a machine;*

Assumption 5 *For the same operation, its processing time is identical in any factory;*

Assumption 6 *To avoid machine blocking, infinite buffers exist between machines in a factory;*

Assumption 7 *A machine may break down during an operation and resume job processing immediately after repair.*

Among these assumptions, Assumptions 1–6 are commonly used in the literature (Naderi and Ruiz, 2010; Wang *et al* (2013c)) to simplify the DPFSP, and Assumption 7 is made to model the process of machine breakdown.

Let $\pi = \{\pi_1, \pi_2, \ldots, \pi_f\}$ represent a feasible solution (schedule) of the addressed DPFSP, where $\pi_k (k = 1, 2, \ldots, f)$ is the job sequence in factory k. The objective of DPFSP in this study is to identify a set of job sequence π to mimimise the makespan, that is, the maximum job completion time in the distributed permutation flowshop. On the basis of the assumptions above, we formulate the DPFSP as follows:

$$\min\left\{C\left(\pi_k^{n_k}, m\right)\right\}, k = 1, 2, \ldots, f \tag{1}$$

Subject to the following constraints:

$$C\left(\pi_k^1, 1\right) = P\left(\pi_k^1, 1\right), k = 1, 2, \ldots, f \tag{2}$$

$$C\left(\pi_k^j, 1\right) = C\left(\pi_k^{j-1}, 1\right) + P\left(\pi_k^j, 1\right), \quad k = 1, 2, \ldots, f; j = 2, 3, \ldots, n_k \tag{3}$$

$$C\left(\pi_k^1, i\right) = C\left(\pi_k^1, i-1\right) + P\left(\pi_k^1, i\right), \quad k = 1, 2, \ldots, f; i = 2, \ldots, m \tag{4}$$

$$C\left(\pi_k^j, i\right) = \max\left\{C\left(\pi_k^{j-1}, i\right), C\left(\pi_k^j, i-1\right)\right\} + P\left(\pi_k^j, i\right), \quad k = 1, 2, \ldots, f; i = 2, 3, \ldots, m; j = 2, 3, \ldots, n_k \tag{5}$$

where

i:	machine index in a factory, $1 \leqslant i \leqslant m$
k:	factory index, $1 \leqslant k \leqslant f$
m:	the number of machines in a factory
f:	the number of factories
n:	the total number of jobs to be processed
n_k:	the number of jobs processed in factory k
π_k^j:	j^{th} job in job sequence $\pi_k = \{\pi_k^1, \pi_k^2, \ldots, \pi_k^{n_k}\}$ in factory k, $1 \leqslant j \leqslant n_k$
$C(\pi_k^j, i)$:	completion time of job j on machine i in factory k
$P(\pi_k^j, i)$:	processing time of job j on machine i in factory k

3. The framework of fuzzy logic-based hybrid EDA (FL-HEDA)

Because of its increasing popularity as a prominent alternative to conventional EAs, EDA has received more attention on a variety of optimisation problems in recent years. In the EDA, the offspring are produced using a probabilistic model rather than the crossover and mutation of GA. Such probabilistic model is constructed from elite individuals of previous population and accordingly can predict promising search areas. The iterative procedure of conventional EDA is presented as follows (Jarboui *et al*, 2009; Wang *et al*, 2012): (1) an initial population is randomly generated; (2) a subpopulation of parent individuals with good performance is selected to build an elite set; (3) the probabilistic model is established based on the statistical information from elite set; (4) new individuals are generated by sampling from the obtained probabilistic model; (5) part/all of the individuals at current population are substituted by the new generated offspring; (6) Steps 2–5 are repeated until a convergence criterion is met.

Since EDA generates offspring entirely from probabilistic models, it may not preserve the population diversity and accordingly tend to get trapped into search stagnation after some generations (Wang *et al*, 2013b). Hybridisation with other EAs, such as GA, may therefore be effective to prevent such search stagnation. Since crossover and mutation of GA perturbs the structure of offspring individuals, they may be incorporated into EDA to improve the population diversity. However, few research works have focused on hybridisation of EDA with GA. Among these proposed hybrid algorithms, ACGA (Chang *et al*, 2008) and eACGA (Chen *et al*, 2012) are two typical ones. Instead of applying an adaptive evolution scheme, both ACGA and eACGA only alternate EDA and GA to generate individuals when a fixed number of generations is reached.

To improve the population diversity, the proposed FL-HEDA integrates EDA with GA for solving the DPFSP under machine breakdown. As shown in Figure 2, two mechanisms differentiate the FL-HEDA from the conventional EDA: the discrete-event simulation model under machine breakdown and the fuzzy logic-based adaptive evolution strategy (FL-AES). To evaluate the expected makespan of offspring individuals, FL-HEDA adopts the discrete-event simulation model to simulate dynamic manufacturing processes in a distributed permutation flowshop. Moreover, instead of using either EDA or GA over a predefined number of generations, FL-HEDA integrates the probabilistic model of EDA with crossover and mutation of GA to generate the new population. In the FL-AES, fuzzy logic is used to preserve the population diversity by dynamically changing the EDA participation ratio R_{EDA}, representing the ratio of offspring produced by EDA.

4. Details of fuzzy logic-based hybrid EDA (FL-HEDA)

To address DPFSPs under machine breakdown efficiently, the proposed FL-HEDA hybridises EDA with GA to produce new population. The details of such hybridisation are presented below.

4.1. Encoding and decoding schemes

Since the permutation-based method has been widely applied to encode solutions for various flowshop scheduling problems (Zhang and Li, 2011; Pan and Ruiz, 2012), it is adopted in this study as an encoding scheme to indicate the job processing order on available machines. For instance, job sequence $\{4, 1, 3, 2\}$ represents a solution to a DPFSP with four jobs. It indicates that Job 4 is first scheduled, then followed by jobs 1, 3, and 2 successively.

To decode a job sequence for the DPFSP, two interdependent decisions, namely job allocation to the suitable factories and job sequence determination in a factory, has to be made. According to the characteristics of distributed permutation flowshop environments, we decode the encoded solutions using an effective earliest completion time (ECT) rule, which has been applied for the DPFSP by Naderi and Ruiz (2010) and Xu *et al* (2014). The proposed ECT rule aims to balance the workload of factories and has three main steps: (1) assign each of the first f jobs in the job sequence to only one factory $F_k (1 \leqslant k \leqslant f)$; (2) allocate each of the remaining jobs to a factory with the minimum completion time; (3) schedule the jobs assigned to a factory in the same order as they appear in the job sequence.

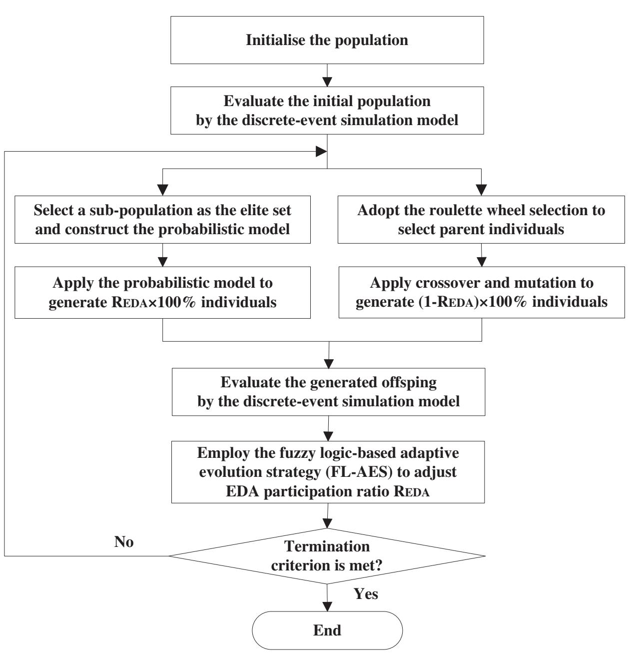

Figure 2 The flowchart of the proposed FL-HEDA.

4.2. Population initialisation

To explore more promising candidate solutions, individuals of the initial population are produced randomly. This population initialisation method is also adopted by Jarboui *et al* (2009) and Tzeng *et al* (2012).

4.3. Population evaluation under machine breakdown

To evaluate the population under machine breakdown, a discrete-event simulator is applied to provide the estimations of expected makespan. In this simulator, the event-driven policy and right-shift schedule repair are adopted to deal with machine breakdown, that is, rescheduling is executed when machine breakdown occurs and the remaining operations are therefore postponed by the downtime. The discrete-event simulator under machine breakdown is detailed as follow, in which both times between failures and repair times for the machines are assumed of exponential distributions.

Notation:

n:	a counter, the number of simulation replications
N_{sim}:	maximum simulation replications
M_{ki}:	machine i in factory k
$MTBF_{ki}$:	mean time between failure of M_{ki}
TBF_{ki}:	time between failure of M_{ki}
$MTTR_{ki}$:	mean time to repair of M_{ki}
TTR_{ki}:	time to repair of M_{ki}
BL_{ki}:	breakdown level of M_{ki}
APT_{ki}:	accumulated processing time of M_{ki}
SM_n:	simulated makespan in the n^{th} simulation replication
SM_{avg}:	average simulated makespan

Algorithm I: The discrete-event simulator algorithm

Step 1: Generate the $MTTR_{ki}$ and BL_{ki} for each machine M_{ki};

Step 2: Compute the $MTBF_{ki}$ for each machine M_{ki} using $MTBF_{ki} = MTTR_{ki}/BL_{ki} - MTTR_{ki}$;

Step 3: Initialise $n = 0$ and set the maximum simulation replications N_{sim};

Step 4: For each machine M_{ki}, let $APT_{ki} = 0$ and set TBF_{ki} to a random number following the exponential distribution with mean $MTBF_{ki}$, that is, $TBF_{ki} = \exp(MTBF_{ki})$;

Step 5: Determine the jobs on each machine according to ECT rule;

Step 6: For each machine M_{ki}, identify the first unprocessed job and add its processing time to APT_{ki};

Step 7: If $APT_{ki} \geqslant TBF_{ki}$, run the machine breakdown algorithm. Otherwise, repeat from Step 6 until all the jobs allocated to a machine have been completed;

Step 8: Compute the simulated makespan M^{n}_{sim}, which is the maximum of job completion times in distributed factories;

Step 9: If $n \geqslant N_{\text{sim}}$, return $SM_{\text{avg}} = \sum_{n=1}^{N_{\text{sim}}} SM_n / N_{\text{sim}}$; otherwise, set $n = n + 1$ and go to Step 4.

Algorithm II: The machine breakdown algorithm

Step 1: For the breakdown machine M_{ki}, set TTR_{ki} to a random number following the exponential distribution with mean $MTTR_{ki}$, that is $TTR_{ki} = \exp(MTTR_{ki})$;

Step 2: For unprocessed jobs on the breakdown machine M_{ki}, compute their completion times, which are the sums of TTR_{ki} and their planned completion times;

Step 3: Let $APT_{ki} = 0$ and set TBF_{ki} to a random number following the exponential distribution with mean $MTBF_{ki}$, that is, $TBF_{ki} = \exp(MTBF_{ki})$.

4.4. Selection operator and probabilistic model

To establish the probabilistic model, EDA extracts the global statistical information from an elite set of a population. In this study, such elite set is directly constructed by selecting the best $\alpha \times 100\%$ ($\alpha \in [0, 1]$) of offspring individuals.

Since the probabilistic model is iteratively applied to produce the offspring, it significantly affects EDA performance (Chen and Chen, 2013). For solving PFSPs with the total flow time criterion, Jarboui *et al* (2009) proposed an effective probabilistic model, in which both job order and similar job blocks of elite individuals are considered. This probabilistic model assigned job j to position k of offspring individuals with the following probability:

$$\pi_{jk} = \frac{\eta_{jk} \times \mu_{j[k-1]}}{\sum_{l \in \Omega_k} \left(\eta_{lk} \times \mu_{l[k-1]} \right)} \tag{6}$$

where η_{jk} indicates the number of times job j appears before or on position k multiplied with a predefined constant δ_1; $\mu_{j[k-1]}$ denotes the number of times job j appears immediately after the job on position $k-1$ multiplied with a predefined constant δ_2; Ω_k represents a set of unassigned jobs that can be allocated to position k.

To address DPFSPs under machine breakdown, the probabilistic model introduced by Jarboui *et al* (2009) may be further improved. According to ECT rule, each of the first f jobs of the job sequence is allocated to only one factory. According to Assumption 5 in Section 2, the same operation takes equal processing time in any factory and therefore job interchange among the first f positions in a job sequence does not change the individual fitness. For instance, consider a DPFSP with $f = 2$, $n = 4$, and $m = 2$. Job sequences [1, 4, 3, 2] and [4, 1, 3, 2] represent two offspring individuals and are converted to two feasible schedules (see Figure 3). As either Job 1 or 4 in this figure is the first job processed at a factory, both [1, 4, 3, 2] and [4, 1, 3, 2] result in the same makespan.

On the basis of the work of Jarboui *et al* (2009), we establish a new probabilistic model, in which job interchanges without affecting the schedule performance are considered. Let f and n denote the numbers of factories and jobs released to the distributed permutation flowshop, respectively. For a specific generation g, job j can be

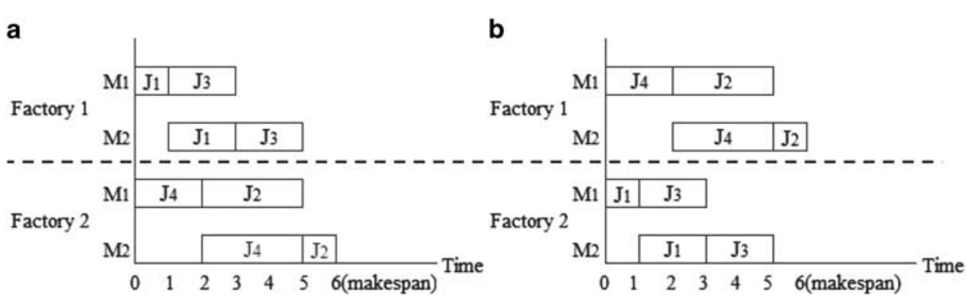

Figure 3 Gantt chart of two schedules. (a) Decoding the job sequence [1, 4, 3, 2]; (b) decoding the job sequence [4, 1, 3, 2].

allocated to position k with the following probability:

$$P_{jk}(g) = \begin{cases} \dfrac{\eta_{jf}}{\sum_{l\in\Omega_k}\eta_{lf}}, & 1\leqslant k\leqslant f \\[2ex] \dfrac{\eta_{jk}\times\mu_{j[k-1]}}{\sum_{l\in\Omega_k}\left(\eta_{lk}\times\mu_{l[k-1]}\right)}, & f<k\leqslant n \end{cases} \tag{7}$$

According to the above probabilistic model, the probabilities for positioning job j in any one of the first f positions, that is, $P_{jk}(g)$ $(1\leqslant k\leqslant f)$, are determined by η_{jf} and accordingly should be identical to each other. Thus, the new probabilistic model considers possible job interchanges shown in Figure 3. In this study, we adopt the probabilistic model in formula (7) to produce $R_{\text{EDA}}\times 100\%$ ($R_{\text{EDA}}\in[0, 1]$) of offspring at each generation.

4.5. Crossover and mutation operators

To avoid early search stagnation, the proposed FL-HEDA employs both crossover and mutation operators of GA to produce a portion of offspring individuals. To explore more promising solution regions, crossover generates the offspring by exchanging partial information from two or more parents (Zhang *et al*, 2013; Qin *et al*, 2014). To preserve the population diversity, mutation produces small perturbations by changing one or more genes in a chromosome (Luo *et al*, 2013).

In this study, FL-HEDA first applies roulette wheel selection to determine the parents for crossover and mutation. Then, two typical genetic operators, that is, the order crossover and the swap mutation, are employed to generate new populations.

- *Order crossover*: The individual is split into three parts according to two random cuts. The second part (middle part) of Child 1 is first directly inherited from Parent 1. The remaining positions of Child 1 are then filled according to the order of jobs appeared in Parent 2. A similar process is applied to generate Child 2. Such crossover process is shown in Figure 4.
- *Swap mutation*: Two randomly selected genes of the parent are interchanged, as illustrated in Figure 5.

For each generation, order crossover is applied to all selected pairs of parents until $(1-R_{\text{EDA}})\times 100\%$ ($R_{\text{EDA}}\in[0, 1]$) of new individuals are generated, and then swap mutation is performed with a mutation rate of p_m.

4.6. Fuzzy logic-based adaptive evolution strategy (FL-AES)

Adaption techniques of EA have recently attracted increasing attention since the effectiveness of EA greatly depends on the parameter setting (Valdez *et al*, 2014). As a commonly used parameter adaption method, fuzzy logic has been successfully applied to tune the parameters of a variety of EAs, such as simulated annealing (SA) heuristic (Jeong *et al*, 2009), GA (Kim *et al*, 2005; Chamnanlor *et al*, 2014), artificial immune system (Chan *et al*, 2013), PSO (Yalaoui *et al*, 2013), hybrid EA (Valdez *et al*, 2011), and so on. To provide an effective hybridisation of EDA and GA, we present a novel fuzzy logic-based adaptive evaluation strategy (FL-AES) to dynamically adjust the EDA participation ratio R_{EDA}, representing the percentage of offspring individuals generated using probabilistic models. Different from previous research work on parameter adaption of EAs, fuzzy logic is first adopted in a hybrid EA to regulate the number of offspring generated by different EAs.

Fuzzy logic, introduced by Zadeh (1973), provides a powerful mathematical framework for performing reasoning in decision-making problems with fuzziness, vagueness and impreciseness (Macías-Escrivá *et al*, 2013; Yalaoui *et al*, 2013). It consists of three steps: fuzzification, decision making, and defuzzification. The general structure of fuzzy logic controller (FLC) in FL-AES is described as Figure 6 and its details are as follows:

- *Input*: The change of EDA participation ratio is determined by the changes of average fitness of the population. Accordingly, FLC inputs include the change of average fitness at current generation (t) and previous generation ($t-1$). Similar to the formulations proposed by Kim *et al* (2003), the inputs are described as:

$$\Delta f_{\text{avg}}(t) = \left(\frac{\sum_{k=1}^{P_S} f(k;\, t)}{P_S} - \frac{\sum_{k=P_S}^{P_S+O_S} f(k;\, t)}{O_S}\right) \tag{8}$$

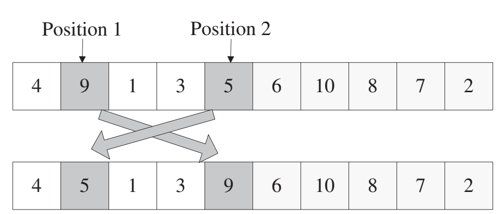

Figure 5 An illustration of swap mutation.

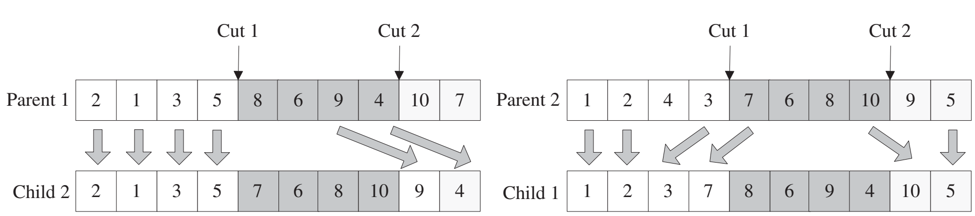

Figure 4 An illustration of order crossover.

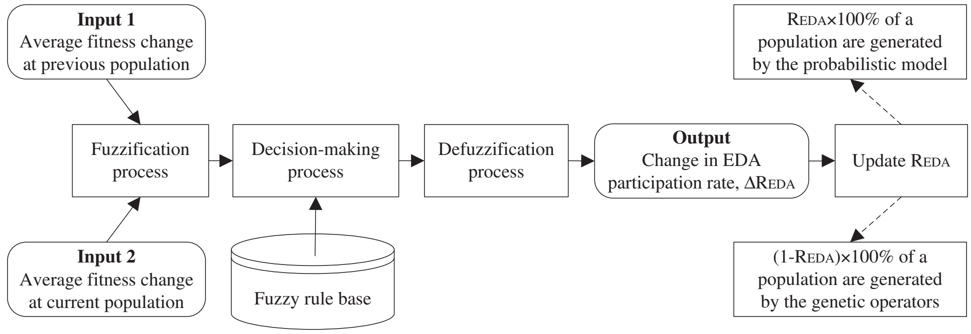

Figure 6 Fuzzy logic control of FL-AES.

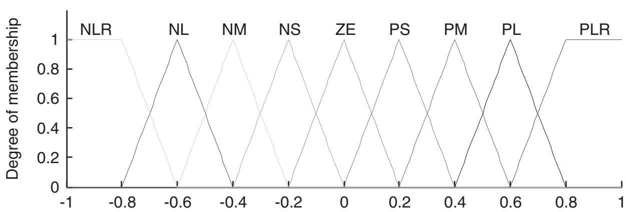

Figure 7 Membership functions of $\Delta f_{\mathrm{avg}}(t-1)$ and $\Delta f_{\mathrm{avg}}(t)$.

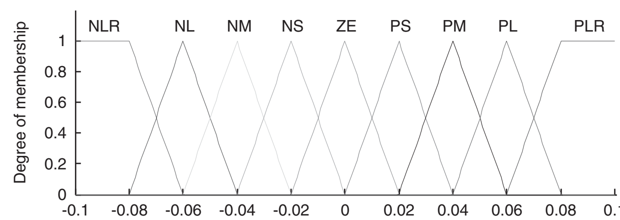

Figure 8 Membership function of ΔR_{EDA}.

$$\Delta f_{\mathrm{avg}}(t-1) = \left(\frac{\sum_{k=1}^{P_S} f(k;\, t-1)}{P_S} - \frac{\sum_{k=P_S}^{P_S+O_S} f(k;\, t-1)}{O_S} \right) \tag{9}$$

where $\Delta f_{\mathrm{avg}}(t)$ represents the change of average fitness at generation *t; f*(k; t) denotes the k^{th} individual's fitness at generation *t;* P_S, and O_S indicate the population size and offspring size, respectively.

- *Output*: The change of EDA participation ratio, that is, ΔR_{EDA}.
- *Input normalisation*: Both $\Delta f_{\mathrm{avg}}(t-1)$ and $\Delta f_{\mathrm{avg}}(t)$ are normalised to the range [1.0, 1.0] as

$$Norm\left(\Delta f_{\mathrm{avg}}(t)\right) = -1 + \frac{\left(\Delta f_{\mathrm{avg}}(t) - \min\left(\Delta f_{\mathrm{avg}}(t)\right)\right) \times 2}{\max\left(\Delta f_{\mathrm{avg}}(t)\right) - \min\left(\Delta f_{\mathrm{avg}}(t)\right)} \tag{10}$$

- *Membership function and fuzzification*: For implementation of fuzzy values into FL-AES, a set of linguistic variables are used, namely Negative Larger (NLR), Negative Large (NL), Negative Medium (NM), Negative Small (NS), Zero (ZE), Positive Small (PS), Positive Medium (PM), Positive Large (PL), and Positive Larger (PLR). Accordingly, the input and output membership functions that associate the linguistic terms to degrees of membership are described in Figures 7 and 8, respectively. According to the input membership function, the normalised values of $\Delta f_{\mathrm{avg}}(t-1)$ and $\Delta f_{\mathrm{avg}}(t)$ are fuzzified into linguistic terms.
- *Fuzzy rule base for decision-making*: After the fuzzyfication process, a commonly used fuzzy rule base for parameter adaption in the literature (Kim *et al*, 2003; Chan *et al*, 2013) is established to determine ΔR_{EDA}. Since each of the two inputs can be described by nine linguistic terms, a total of $9 \times 9 = 81$ fuzzy rules are developed to cover all possible input combinations. Table 1 illustrates the rules inside the base.

Table 1 Fuzzy decision table for the change of EDA participation ratio (ΔR_{EDA})

$\Delta f_{\mathrm{avg}}(t)$	$\Delta f_{\mathrm{avg}}(t-1)$								
	NLR	*NL*	*NM*	*NS*	*ZE*	*PS*	*PM*	*PL*	*PLR*
NLR	NLR	NL	NL	NM	NM	NS	NS	ZE	ZE
NL	NL	NL	NM	NM	NS	NS	ZE	ZE	PS
NM	NL	NM	NM	NS	NS	ZE	ZE	PS	PS
NS	NM	NM	NS	NS	ZE	ZE	PS	PS	PM
ZE	NM	NS	NS	ZE	ZE	PS	PS	PM	PM
PS	NS	NS	ZE	ZE	PS	PS	PM	PM	PL
PM	NS	ZE	ZE	PS	PS	PM	PM	PL	PL
PL	ZE	ZE	PS	PS	PM	PM	PL	PL	PLR
PLR	ZE	PS	PS	PM	PM	PL	PL	PLR	PLR

- *Defuzzification*: The fuzzy value of ΔR_{EDA} obtained by the decision-making process is further defuzzified using the gravity centre method, which is described as:

$$\Delta R_{\mathrm{EDA}} = \frac{\int_{x=-0.1}^{x=0.1} x \times \mu(x)dx}{\int_{x=-0.1}^{x=0.1} \mu(x)dx} \tag{11}$$

where x represents the horizontal axis of Figure 8 and $\mu(x)$ denotes the maximum value of each linguistic term.

Integrated with FLC introduced above, the procedure of FL-AES is described as follows:

Algorithm III: FL-AES

Step 1: Compute the changes of average fitness of two continuous generations ($t-1$ and t), that is, $\Delta f_{\text{avg}}(t-1)$ and $\Delta f_{\text{avg}}(t)$;

Step 2: Normalise $\Delta f_{\text{avg}}(t-1)$ and $\Delta f_{\text{avg}}(t)$, and then fuzzify them into suitable linguistic terms;

Step 3: Perform a fuzzy reasoning according to the fuzzy rules shown in Table 1 and obtain ΔR_{EDA} by defuzzification;

Step 4: Obtain R_{EDA} at generation t using one of the two following strategies, in which β is a scaling factor to adjust the range of ΔR_{EDA}.

(a) $R_{\text{EDA}}(t) = \min\{R_{\text{EDA}}(t-1) + \beta \times \Delta R_{\text{EDA}}, 1\}$, if $\Delta R_{\text{EDA}} \geqslant 0$

(b) $R_{\text{EDA}}(t) = \max\{R_{\text{EDA}}(t-1) + \beta \times \Delta R_{\text{EDA}}, 0\}$, if $\Delta R_{\text{EDA}} < 0$

4.7. Stopping criterion

To address the flowshop scheduling problems, four different criteria have been applied to terminate EDA, including the maximum number of generations (Jarboui *et al*, 2009), the maximum number of evaluations (Wang *et al* (2013d)), and the maximum computation time (Pan and Ruiz, 2012; Tzeng *et al*, 2012). In this study, FL-HEDA terminates when reaching the predefined number of generations.

4.8. Implementation of FL-HEDA for DPFSP

Having discussed all components of FL-HEDA, the procedure of FL-HEDA for solving the DPFSP under machine breakdown is presented below:

Notations:

- N_{sim}: maximum simulation replications
- P_s: population size
- P_m: mutation rate
- α: population ratio to update the probabilistic model
- β: scaling factor to adjust the range of ΔR_{EDA}
- R_{EDA}: EDA participation ratio
- S_c: best solution at current generation
- S_b: best solution over generations
- *CurGen*: index of current generation
- *MaxGen*: total number of generations

Algorithm IV: FL-HEDA

Step 1: Initialise key parameters of FL-HEDA, including N_{sim}, P_s, *MaxGen*, α, β, and P_m. Set $R_{\text{EDA}} = 50\%$ and $CurGen = 1$;

Step 2: Produce a random initial population with P_s individuals and estimate their performance by a discrete-event simulator;

Step 3: Identify the best individual of current population S_b;

Step 4: Choose the best $\alpha \times P_s$ individuals to construct an elite set Π_e;

Step 5: Use formula (7) to estimate a probabilistic model based on Π_e;

Step 6: Generate $R_{\text{EDA}} \times P_s$ offspring using the obtained probabilistic model;

Step 7: Generate $(1 - R_{\text{EDA}}) \times P_s$ offspring using order crossover and swap mutation;

Step 8: Apply the discrete-event simulator to evaluate the current population under machine breakdown;

Step 9: Identify the best individual S_c of current population and update $S_b = S_c$ if necessary;

Step 10: Adjust R_{EDA} using FL-AES;

Step 11: Terminate the algorithm and return S_b if stopping criterion $CurGen \geqslant MaxGen$ is met; otherwise, set $CurGen = CurGen + 1$ and return to Step 4.

5. Computational results and discussion

5.1 Design of experiments

The FL-HEDA is coded in Java programming language and runs on a regular PC workstation with Intel® Core™ i3 2.10 GHz processor and 4 GB memory. To validate the effectiveness of the proposed FL-HEDA, we design and implement two experiments: one to identify appropriate parameter values of FL-HEDA, and the other to evaluate the performance of FL-HEDA under machine breakdown.

In the above experiments, job processing times are assumed to be uniformly distributed with range $[0, 20]$. The average of processing times, that is, PT_{avg}, is therefore 10 time units. Furthermore, to mimic the production process in dynamic manufacturing environments, we model machine breakdown using *MTTR* and *BL*, which are uniformly generated with ranges $[PT_{\text{avg}}, 2 \times PT_{\text{avg}}]$ and $[0.01, 0.1]$ for each machine respectively. In this study, both the times between failures and the repair times follow the exponential distributions with mean values of *MTBF* and *MTTR*.

5.2. Experiment I: Parameter tuning

The FL-HEDA depends on five key parameters, namely P_s (the population size), *MaxGen* (the number of generations), α (population ratio to update the probabilistic model), β (the scaling factor to adjust the range of ΔR_{EDA}), and p_m (mutation rate). To obtain appropriate values of these five parameters, the Taguchi method (1986) is implemented on a moderate-sized DPFSP ($n = 60$, $m = 10$, and $f = 3$) to analyse different parameter settings of FL-HEDA. To investigate the effect of factors on response variables, this method enables the drastic reduction of the number of experiments compared with a full factorial experimental design (Naderi *et al*, 2009). In this study, these five key parameters of FL-HEDA and the average makespan under machine breakdown are the factors and the response variable in the Taguchi experiments, respectively.

Table 2 Factors and factor levels of FL-HEDA

Factor	*Number of levels*	*Level 1*	*Level 2*	*Level 3*	*Level 4*
P_s	4	100	200	300	400
MaxGen	4	100	200	300	400
α	4	0.10	0.15	0.20	0.25
β	4	0.60	0.80	1.00	1.20
p_m	4	0.10	0.15	0.20	0.25

Table 3 Orthogonal array L_{16} (4^5) of FL-HEDA

Trial	*Factor*					*RV*
	P_s	*MaxGen*	α	β	p_m	
1	1	1	1	1	1	380.8
2	1	2	2	2	2	375.6
3	1	3	3	3	3	376.0
4	1	4	4	4	4	379.9
5	2	1	2	3	4	375.5
6	2	2	1	4	3	374.2
7	2	3	4	1	2	376.1
8	2	4	3	2	1	372.6
9	3	1	3	4	2	374.7
10	3	2	4	3	1	374.7
11	3	3	1	2	4	372.6
12	3	4	2	1	3	371.4
13	4	1	4	2	3	377.6
14	4	2	3	1	4	374.1
15	4	3	2	4	1	370.5
16	4	4	1	3	2	370.6

Table 2 presents the factors of FL-HEDA and their different levels considered in the Taguchi method. Using MINITAB 16, an orthogonal array L_{16} (4^5) presented in Table 3 is established for parameter tuning. Instead of performing $4^5 = 1024$ experiments using the full factorial approach, Taguchi method requires only 16 experiments to determine the near optimum value of each factor. For each Taguchi experiment shown in Table 3, FL-HEDA first runs 20 times to address the selected moderate-sized DPFSP. The average makespan under machine breakdown is subsequently determined as the value of response variable (RV), which is also included in Table 3. According to the Taguchi experimental results, Figure 9 presents RV value of each factor level. In this figure, *x*-axes represent the five factors considered in the Taguchi experiments shown in Table 3, and *y*-axes indicate the average makespan of each factor level. Since this study aims to minimise the makespan of DPFSP, FL-HEDA may has better performance when these five factors are set to the values with the minimum average makespan.

To evaluate the relative significance of these five factors, delta, representing the difference of maximum and minimum values of average RV for a specific factor, is applied in this study. Table 4 presents the delta and ranks of these five factors. The lower rank of a factor indicates its more significant influence on the performance of FL-HEDA.

From the ranks in Table 4, it is obvious that P_s is the most significant one among the five parameters. Although a large P_s would improve solutions by performing a better sampling of the promising search area, large computation cost of evaluating the population under machine breakdown has to be required. Since Figure 9 implies that no significant improvement of solutions occurs when P_s is too large, the parameter P_s of 300 is used in this study to avoid long computation time. Besides, α ranks the second and is also crucial to the performance of

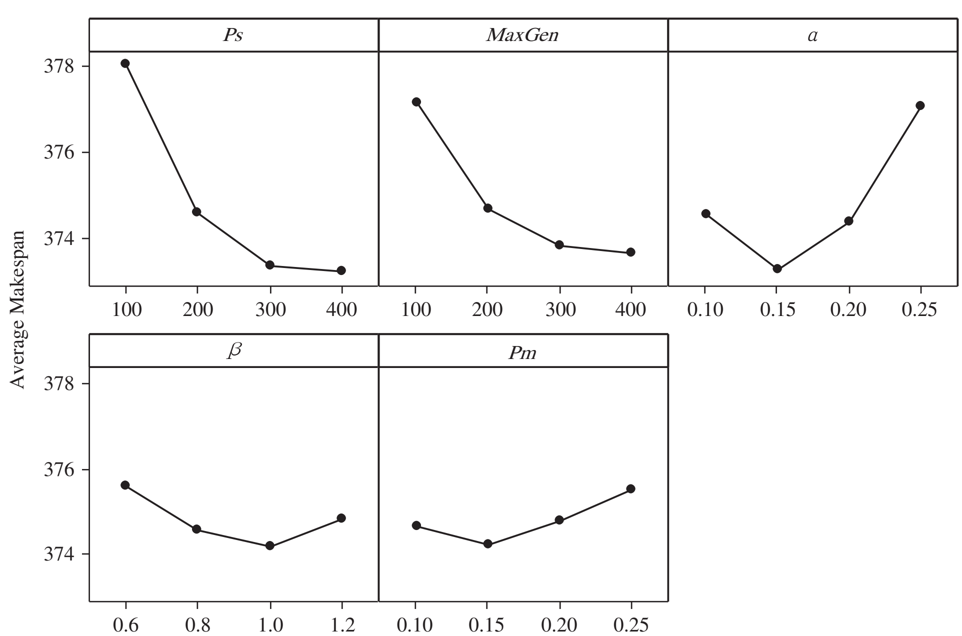

Figure 9 Factor level trend of FL-HEDA.

FL-HEDA. A moderate value of α may provide a better probabilistic model to describe the promising solutions and is accordingly selected as 0.15. In addition, *MaxGen* is the third significant parameter and its value is set to 300 when considering the amount of computation time. Furthermore, the values of other parameters, that is, β and p_m, can be directly identified according to Figure 9. On the basis of the discussion above, the values of key parameters in FL-HEDA are therefore determined as follows: $P_s = 300$, $MaxGen = 300$, $\alpha = 0.15$, $\beta = 1.0$, and $p_m = 0.15$.

Table 4 Average response at each factor level

Level	*Factor*				
	P_s	*MaxGen*	α	β	p_m
1	378.1	377.1	374.5	375.6	374.7
2	374.6	374.6	373.3	374.6	374.3
3	373.4	373.8	374.4	374.2	374.8
4	373.2	373.6	377.1	374.8	375.5
Delta (max–min)	4.9	3.5	3.8	1.4	1.3
Rank	1	3	2	4	5

5.3. Experiment II: Performance evaluation of FL-HEDA

In real-world manufacturing systems, machine breakdown may inevitably result in deviations from the planned schedules and eventually degrade the performance of scheduling algorithms. To evaluate the performance of FL-HEDA under machine breakdown, we compared it with some other meta-heuristics based on a test-bed containing 27 DPFSPs, where $n = \{50, 100, 150\}$, $m = \{5, 10, 15\}$, and $f = \{2, 3, 4\}$. For each DPFSP in the test-bed, 10 instances are considered, in which job processing times follow a uniform distribution in the range from 1 to 20 time units. Therefore, the total number of instances in this experiment reaches 270.

For each of DPFSP instances, 20 independent replications are conducted to evaluate different meta-heuristics. Their performance is measured by the minimum relative percentage deviation (MRPD) and the average relative percentage deviation (ARPD), which are described below respectively:

$$MRPD = \frac{1}{10}\sum_{r=1}^{10} \frac{\left(BS^r_{\text{alg}} - S^r_{\text{best}}\right) \times 100}{S^r_{\text{best}}} \tag{12}$$

Table 5 Comparison of FL-HEDA and HEDAs with constant R_{EDA}

Problem number	*Number of jobs × number of machine × number of factories*	*FL-HEDA*		*HEDA with constant R_{EDA})*					
				$R_{EDA} = 25\%$		*$R_{\text{EDA}} = 50\%$*		*$R_{\text{EDA}} = 75\%$*	
		MRPD	*ARPD*	*MRPD*	*ARPD*	*MRPD*	*ARPD*	*MRPD*	*ARPD*
1	50 × 5 × 2	0.00	0.10	0.38	0.53	0.84	1.14	1.23	1.49
2	50 × 5 × 3	0.00	0.00	0.30	0.42	1.08	1.41	1.00	1.55
3	50 × 5 × 4	0.14	0.27	0.11	0.39	0.73	0.95	1.13	1.33
4	50 × 10 × 2	0.10	0.21	0.50	0.68	0.86	1.16	1.45	1.75
5	50 × 10 × 3	0.00	0.23	0.36	0.52	1.04	1.33	1.26	1.50
6	50 × 10 × 4	0.09	0.18	0.41	0.61	0.77	1.02	1.35	1.61
7	50 × 15 × 2	0.16	0.25	0.52	0.71	0.90	1.18	1.33	1.71
8	50 × 15 × 3	0.10	0.24	0.46	0.60	1.05	1.37	1.25	1.52
9	50 × 15 × 4	0.25	0.31	0.20	0.48	0.88	1.21	1.28	1.58
10	100 × 5 × 2	0.13	0.22	0.64	0.85	1.10	1.42	1.60	1.89
11	100 × 5 × 3	0.19	0.32	0.57	0.76	1.31	1.49	1.24	1.72
12	100 × 5 × 4	0.12	0.23	0.39	0.58	0.82	1.11	1.38	1.63
13	100 × 10 × 2	0.18	0.24	0.43	0.61	0.92	1.21	1.21	1.47
14	100 × 10 × 3	0.08	0.19	0.56	0.77	0.82	1.17	1.30	1.61
15	100 × 10 × 4	0.19	0.27	0.32	0.51	1.09	1.39	1.10	1.35
16	100 × 15 × 2	0.27	0.37	0.60	0.76	1.13	1.46	1.59	1.89
17	100 × 15 × 3	0.15	0.21	0.66	0.86	1.11	1.38	1.41	1.66
18	100 × 15 × 4	0.26	0.30	0.45	0.62	0.89	1.14	1.25	1.51
19	150 × 5 × 2	0.17	0.29	0.49	0.71	0.93	1.22	1.52	1.84
20	150 × 5 × 3	0.13	0.21	0.41	0.61	0.89	1.15	1.51	1.75
21	150 × 5 × 4	0.22	0.33	0.59	0.79	1.07	1.38	1.31	1.56
22	150 × 10 × 2	0.25	0.37	0.60	0.81	1.14	1.49	1.58	1.95
23	150 × 10 × 3	0.11	0.19	0.54	0.75	1.03	1.37	1.52	1.84
24	150 × 10 × 4	0.16	0.23	0.38	0.57	0.91	1.18	1.31	1.61
25	150 × 15 × 2	0.25	0.39	0.62	0.91	1.23	1.53	1.77	2.11
26	150 × 15 × 3	0.29	0.43	0.68	0.98	1.15	1.44	1.67	1.98
27	150 × 15 × 4	0.24	0.34	0.57	0.75	1.04	1.38	1.57	1.87
Average		0.16	0.26	0.47	0.67	0.99	1.28	1.37	1.68

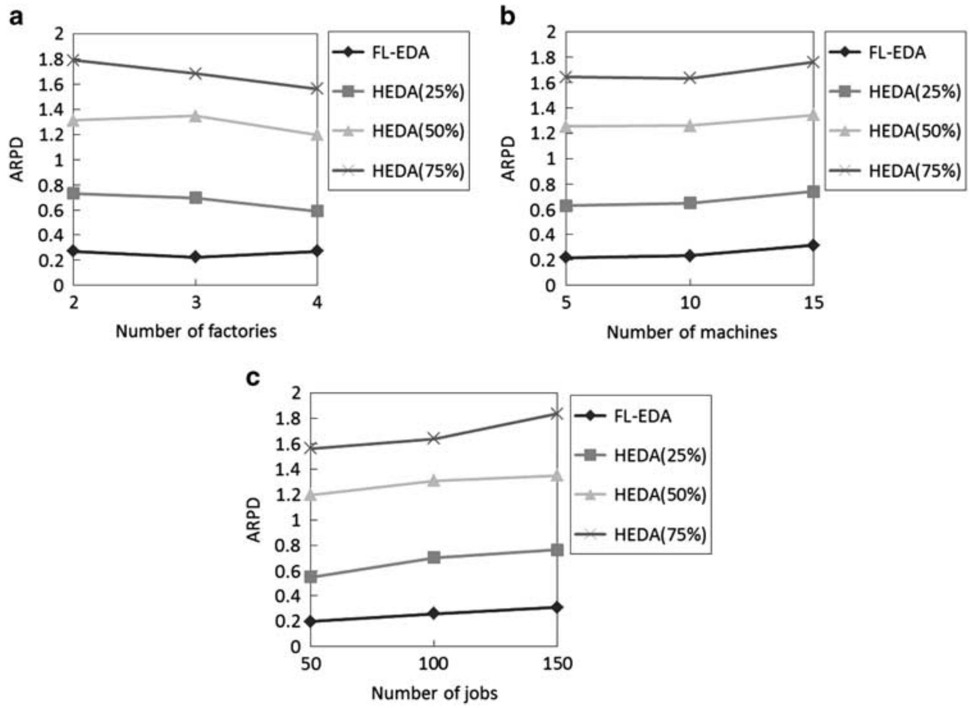

Figure 10 The trend of ARPD for FL-HEDA and HEDAs with constant R_{EDA}. (a) ARPD against the number of factories; (b) ARPD against the number of machines; (c) ARPD against the number of jobs.

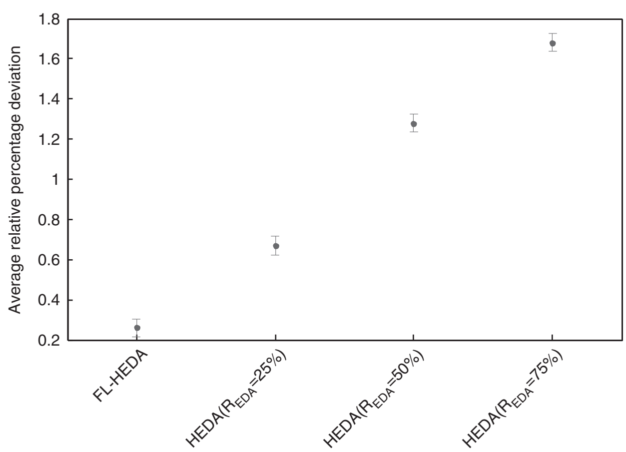

Figure 11 The mean plot with LSD intervals at a 95% confidence level for FL-HEDA and HEDAs with constant R_{EDA}.

$$ARPD = \frac{1}{10}\sum_{r=1}^{10}\frac{\left(AS_{\mathrm{alg}}^{r} - S_{\mathrm{best}}^{r}\right)\times 100}{S_{\mathrm{best}}^{r}} \tag{13}$$

where BS_{alg}^{r} and AS_{alg}^{r} represent makespans of the best and average solutions found by a specific scheduling algorithm for instance r, and S_{best}^{r} indicates the minimum makespan found by all scheduling algorithms for instance r. According to the above two performance measures, an algorithm with lower MRPD and APRD may provide better solutions under machine breakdown.

5.3.1. Comparison of FL-HEDA and hybrid EDAs with constant EDA participation ratio. The uniqueness of FL-HEDA lies in applying FL-AES to guide the generation of offspring. Instead of using a constant EDA participation ratio R_{EDA}, FL-AES adopts fuzzy logic to dynamically adjust R_{EDA} according to the changes of average fitness at previous and current generations. To validate the effectiveness of FL-AES, FL-HEDA is compared with a number of hybrid EDAs (HEDA), which are similar to FL-HEDA except employing a constant R_{EDA} to produce the offspring.

Table 5 presents the experiment results of FL-HEDA and HEDAs with three typical constant R_{EDA} under machine breakdown. As shown in Table 5, it is clear that FL-HEDA provides the best performance in terms of ARPD and MRPD for most test problems. Such good performance of FL-HEDA therefore indicates the superiority of FL-AES.

To visually compare the performance of FL-HEDA and HEDAs with constant R_{EDA}, Figure 10 illustrates the general trend of ARPD with different numbers of factories, machines, and jobs. It is clear that FL-HEDA performs the best among the compared algorithms for all sizes of factories, machines, and jobs.

To further analyse the experiment results, the analysis of variance (ANOVA) and the least significant difference (LSD) tests are conducted. Figure 11 depicts the mean plot with LSD

Table 6 Comparison of FL-HEDA, EDA, GA, HEDA-AO, and PSO_{VNS}

Problem number	*Number of jobs × number of machine × number of factories*	*FL-HEDA*		*EDA*		*GA*		*HEDA-AO*		PSO_{VNS}	
		MRPD	*ARPD*	*MRPD*	*ARPD*	*MRPD*	*ARPD*	*MRPD*	*ARPD*	*MRPD*	*ARPD*
1	50×5×2	0.05	0.06	1.45	1.80	0.63	0.80	0.40	0.55	0.19	0.27
2	50×5×3	0.07	0.11	1.31	1.59	0.42	0.74	0.55	0.74	0.32	0.45
3	50×5×4	0.00	0.00	1.56	1.89	0.68	0.94	0.58	0.81	0.36	0.51
4	50×10×2	0.12	0.17	1.61	1.98	0.46	0.70	0.45	0.54	0.21	0.29
5	50×10×3	0.33	0.39	1.84	2.20	0.88	1.22	0.57	0.69	0.37	0.42
6	50×10×4	0.29	0.35	1.39	1.76	0.67	0.94	0.22	0.41	0.26	0.44
7	50×15×2	0.22	0.27	1.73	2.13	0.83	1.12	0.73	0.95	0.53	0.66
8	50×15×3	0.13	0.20	1.51	1.91	0.69	0.98	0.57	0.78	0.36	0.53
9	50×15×4	0.05	0.11	1.36	1.76	0.57	0.89	0.43	0.63	0.21	0.34
10	100×5×2	0.14	0.19	1.72	2.07	0.74	1.06	0.67	0.89	0.41	0.59
11	100×5×3	0.23	0.33	1.88	2.23	1.02	1.37	0.87	1.11	0.65	0.86
12	100×5×4	0.12	0.17	1.54	1.92	0.86	1.22	0.62	0.81	0.37	0.52
13	100×10×2	0.11	0.21	1.35	1.71	0.71	0.96	0.46	0.68	0.24	0.42
14	100×10×3	0.08	0.12	1.58	1.94	0.53	0.73	0.68	0.89	0.48	0.63
15	100×10×4	0.24	0.29	1.86	2.18	1.03	1.34	0.56	0.74	0.36	0.47
16	100×15×2	0.27	0.35	1.95	2.31	0.91	1.19	0.47	0.65	0.27	0.34
17	100×15×3	0.08	0.13	1.55	1.89	0.76	1.03	0.39	0.54	0.19	0.27
18	100×15×4	0.24	0.31	1.37	1.72	0.53	0.81	0.57	0.78	0.35	0.48
19	150×5×2	0.29	0.34	1.87	2.19	1.01	1.33	0.74	0.98	0.52	0.71
20	150×5×3	0.12	0.16	1.51	1.93	0.90	1.20	0.58	0.77	0.38	0.49
21	150×5×4	0.21	0.28	1.69	2.09	0.67	0.95	0.63	0.83	0.41	0.56
22	150×10×2	0.30	0.35	1.82	2.21	1.02	1.29	0.90	1.19	0.66	0.92
23	150×10×3	0.14	0.17	1.39	1.77	0.78	1.05	0.55	0.78	0.34	0.53
24	150×10×4	0.10	0.14	1.56	1.93	0.62	0.89	0.37	0.56	0.15	0.28
25	150×15×2	0.37	0.45	1.78	2.24	1.15	1.54	0.93	1.18	0.74	0.93
26	150×15×3	0.26	0.37	1.93	2.36	0.94	1.28	0.65	0.97	0.42	0.70
27	150×15×4	0.10	0.18	1.74	2.15	1.02	1.38	0.67	0.83	0.44	0.37
Average		0.17	0.23	1.62	1.99	0.78	1.07	0.59	0.79	0.38	0.52

intervals at a 95% confidence level for FL-HEDA and HEDAs with constant R_{EDA}. As it can be observed from this figure, FL-HEDA yields statistically better results than other algorithms.

5.3.2 Comparisons of FL-HEDA, EDA, GA, and HEDA with alternative operators. FL-HEDA is characterised by the hybirdisation of EDA and GA for population generation. Therefore, to evaluate the effectiveness of such hybridisation, we compare FL-HEDA with EDA, GA, and HEDA with alternative operators (HEDA-AO), and PSO_{VNS} (Tasgetiren *et al*, 2007) under machine breakdown, respectively. The descriptions of these four scheduling algorithms are briefly summarised below:

- *EDA*: This algorithm is the same as FL-HEDA with $R_{EDA}=100\%$ and it therefore only adopts the probabilistic model to construct solutions.
- *GA*: This is a standard GA. Similar to FL-HEDA, it randomly initialises the population and applies both the order crossover and the swap mutation operators to generate new populations. To have a relatively fair comparison with FL-HEDA, the crossover rate and mutation rate of GA are set to 1 and 0.15, respectively.
- *HEDA-AO*: This algorithm generates a new population using either the probabilistic model of EDA or genetic operators of GA. Similar to the decision process of eACGA (Chen *et al*, 2012), the selection of an appropriate approach is guided by two parameters, namely *startingGen* and *interval*. The probabilistic model of EDA is only applied to generate the offspring individuals when $CurGen > startingGen$ and $CurGen\%interval = 0$. In this study, these two parameters are set to the same values as those of eACGA (Chen *et al*, 2012), that is, $startingGen = 0.5 \times MaxGen$ and $interval = 0.02 \times MaxGen$.
- PSO_{VNS}: This algorithm, proposed by Tasgetiren *et al* (2007), integrates PSO with variable neighbourhood search (VNS) to address the PFSP in a static environment. To model machine breakdown in real-world manufacturing systems, we incorporate the discrete-event simulator into PSO_{VNS} to estimate the schedule performance.

In this experiment, these compared scheduling algorithms adopt the same N_{sim}, P_s, and *MaxGen*. The experimental results under machine breakdown are shown in Table 6. From this table, we observe the following: (1) FL-HEDA outperforms either GA or EDA in terms of MRPD and ARPD for most test problems. The superiority of FL-HEDA arises from generating the offspring of

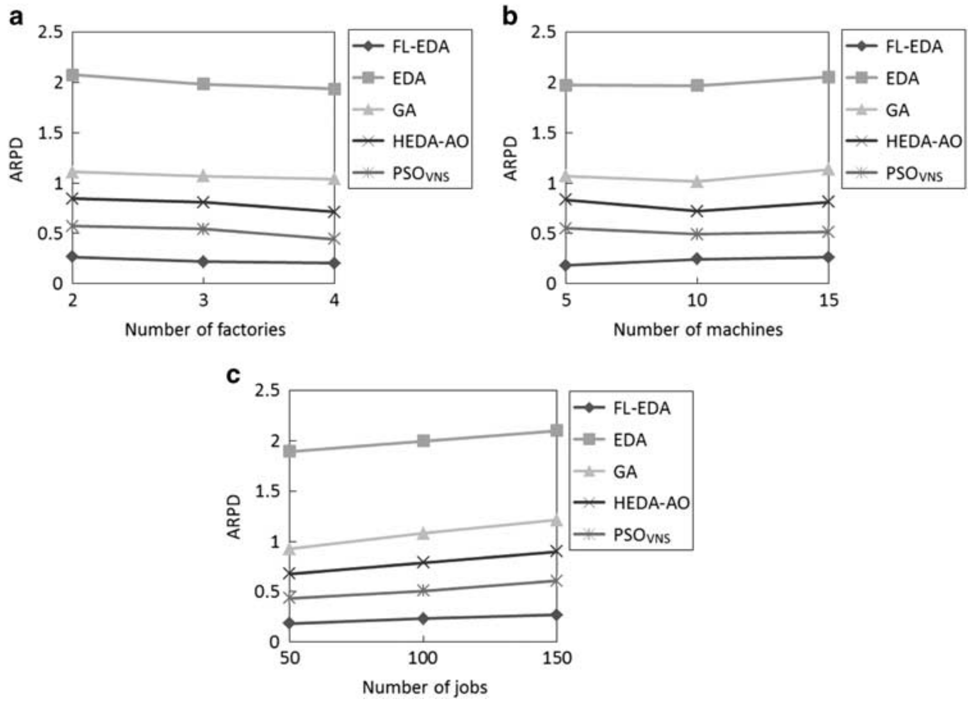

Figure 12 The trend of ARPD for FL-HEDA, EDA, GA, HEDA-AO, and PSO_{VNS}. (a) ARPD against the number of factories; (b) ARPD against the number of machines; (c) ARPD against the number of jobs.

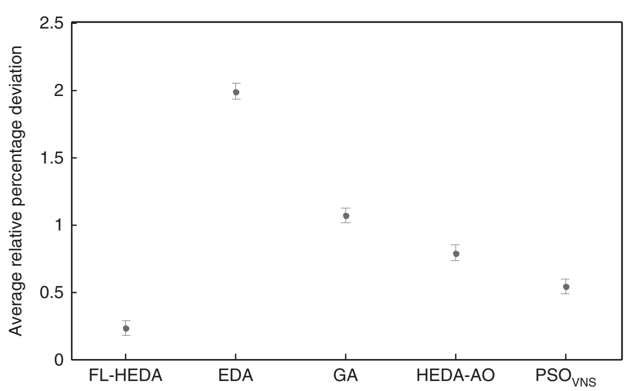

Figure 13 The mean plot with LSD intervals at a 95% confidence level for FL-HEDA, EDA, GA, HEDA-AO, and PSO_{VNS}.

a population using both the probabilistic model of EDA and genetic operators of GA. (2) Compared with HEDA-AO, FL-HEDA gives a smaller MRPD and ARPD, which indicates that FL-AES hybridises EDA and GA in a more effective way than simply switching from one to another when addressing the DPFSP under machine breakdown. (3) FL-HEDA provides better results than PSO_{VNS}, which indicates the effectiveness of hybridisation of EDA and GA in FL-HEDA.

Figure 12 presents the general trend of ARPD with different numbers of factories, machines, and jobs for these five scheduling algorithms above. This figure indicates that FL-HEDA performs the best among the compared algorithms. Furthermore, Figure 13 depicts the mean plot with LSD intervals at a 95% confidence level for these scheduling algorithms. It is obvious that FL-HEDA statistically performs better than the other scheduling algorithms.

6. Conclusion

In this paper, an effective fuzzy logic-based hybrid EDA (FL-HEDA) is presented to minimise the makespan of the distributed permutation flowshop scheduling problems (DPFSP) under machine breakdown. Different from the conventional EDA, FL-HEDA hybridises the probabilistic model with crossover and mutation operators to generate new populations. To enhance the population diversity, a fuzzy logic-based adaptive evolution strategy (FL-AES) is used to dynamically change the ratio of offspring generated by the probabilistic model. Moreover, to evaluate the offspring in a real-world manufacturing environment, a discrete-event simulator is incorporated into FL-HEDA to simulate the production process under machine breakdown. The performance of FL-HEDA has been evaluated based on a test-bed of 27 DPFSPs. We compared the proposed FL-HEDA with hybrid EDAs with constant R_{EDA}, hybrid EDA with alternative operators, EDA, and GA under machine breakdown. The experiment results indicate the effectiveness of the hybridisation of EDA with GA in the FL-HEDA.

The majority of the literature on flowshop scheduling commonly assumes infinite buffers between two consecutive machines. The capacities of such buffers, however, are limited in real-world manufacturing systems, which may delay job processing on the previous machine. Motivated by a desire to make the proposed FL-HEDA more practical, future research work can focus on extending it to address the DPFSP with limited buffers. Furthermore, machine breakdown is usually modelled by two parameters in the literature, namely the *MTTR* and the machine *BL*. Thus, another promising research area is to investigate how different values of *MTTR* and *BL* affect the performance of FL-HEDA. In addition, the effectiveness of FL-AES is related to the fuzzy rule base and it therefore would be of great academic interest to refine the fuzzy rules for improving the performance of FL-HEDA.

Acknowledgements—We would like to thank the anonymous referees for their constructive and pertinent comments. This research is partially supported by Key Program from National Natural Science Foundation of China (No. 71131004 and No. 71231007), National Science Foundation of China (No. 71301124), Humanities and Social Sciences Foundation of the Ministry of Education of China (No.13YJC630165), Fundamental Research Funds for the Central Universities (No.2012GSP026), Macau University of Science and Technology (No. 0237), and Macau Science and Technology Development Fund (No. 066/2013/A2).

References

Adibi MA, Zandieh M and Amiri M (2010). Multi-objective scheduling of dynamic job shop using variable neighborhood search. *Expert Systems with Applications* **37**(1): 282–287.

Aytug H, Lawley MA, McKay K, Mohan S and Uzsoy R (2005). Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research* **161**(1): 86–110.

Ceberio J, Irurozki E, Mendiburu A and Lozano JA (2014). A distance-based ranking model estimation of distribution algorithm for the flowshop scheduling problem. *Evolutionary Computation, IEEE Transactions* **18**(2): 286–300.

Chamnanlor C, Sethanan K, Chien CF and Gen M (2014). Re-entrant flow shop scheduling problem with time windows using hybrid genetic algorithm based on auto-tuning strategy. *International Journal of Production Research* **52**(9): 2612–2629.

Chan FT, Prakash A, Ma HL and Wong CS (2013). A hybrid tabu sample-sort simulated annealing approach for solving distributed scheduling problem. *International Journal of Production Research* **51**(9): 2602–2619.

Chan FT, Prakash A and Mishra N (2013). Priority-based scheduling in flexible system using AIS with FLC approach. *International Journal of Production Research* **51**(16): 4880–4895.

Chan HK and Chung SH (2013). Optimisation approaches for distributed scheduling problems. *International Journal of Production Research* **51**(9): 2571–2577.

Chang PC, Chen SS and Fan CY (2008). Mining gene structures to inject artificial chromosomes for genetic algorithm in single machine scheduling problems. *Applied Soft Computing* **8**(1): 767–777.

Chen WJ (2006). Minimizing total flow time in the single-machine scheduling problem with periodic maintenance. *Journal of the Operational Research Society* **57**(4): 410–415.

Chen YM, Chen MC, Chang PC and Chen SH (2012). Extended artificial chromosomes genetic algorithm for permutation flowshop scheduling problems. *Computers & Industrial Engineering* **62**(2): 536–545.

Chen SH and Chen MC (2013). Addressing the advantages of using ensemble probabilistic models in estimation of distribution algorithms for scheduling problems. *International Journal of Production Economics* **141**(1): 24–33.

De Giovanni L and Pezzella F (2010). An improved genetic algorithm for the distributed and flexible job-shop scheduling problem. *European journal of operational research* **200**(2): 395–408.

Gao J and Chen R (2011). A hybrid genetic algorithm for the distributed permutation flowshop scheduling problem. *International Journal of Computational Intelligence Systems* **4**(4): 497–508.

Gao J, Chen R and Deng W (2013). An efficient tabu search algorithm for the distributed permutation flowshop scheduling problem. *International Journal of Production Research* **51**(3): 641–651.

Garey MR, Johnson DS and Sethi R (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research* **1**(2): 117–129.

Hatami S, Ruiz R and Andrés-Romano C (2013). The distributed assembly permutation flowshop scheduling problem. *International Journal of Production Research* **51**(17): 5292–5308.

Jarboui B, Eddaly M and Siarry P (2009). An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems. *Computers & Operations Research* **36**(9): 2638–2646.

Jeong SJ, Kim KS and Lee YH (2009). The efficient search method of simulated annealing using fuzzy logic controller. *Expert Systems with Applications* **36**(3): 7099–7103.

Kim KW, Gen M and Yamazaki G (2003). Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling. *Applied Soft Computing* **2**(3): 174–188.

Kim K, Yun Y, Yoon J, Gen M and Yamazaki G (2005). Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling. *Computers in Industry* **56**(2): 143–160.

Leitão P (2009). Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence* **22**(7): 979–991.

Lin SW, Ying KC and Huang CY (2013). Minimising makespan in distributed permutation flowshops using a modified iterated greedy algorithm. *International Journal of Production Research* **51**(16): 5029–5038.

Liu H, Gao L and Pan Q (2011). A hybrid particle swarm optimization with estimation of distribution algorithm for solving permutation flowshop scheduling problem. *Expert Systems with Applications* **38**(4): 4348–4360.

Luo H, Du B, Huang GQ, Chen H and Li X (2013). Hybrid flow shop scheduling considering machine electricity consumption cost. *International Journal of Production Economics* **146**(2): 423–439.

Luo H, Huang GQ, Shi Y, Qu T and Zhang YF (2012). Hybrid flowshop scheduling with family setup time and inconsistent family formation. *International Journal of Production Research* **50**(6): 1457–1475.

Macías-Escrivá FD, Haber R, del Toro R and Hernandez V (2013). Self-adaptive systems: A survey of current approaches, research challenges and applications. *Expert Systems with Applications* **40**(18): 7267–7279.

Mühlenbein H and Paass G (1996). From recombination of genes to the estimation of distributions I: Binary parameters. *Lecture Notes in Computer Science* **1141**: 178–187, http://link.springer.com/chapter/10.1007/3-540-61723-X_982.

Naderi B, Zandieh M, Khaleghi Ghoshe Balagh A and Roshanaei V (2009). 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): 9625–9633.

Naderi B and Ruiz R (2010). The distributed permutation flowshop scheduling problem. *Computers & Operations Research* **37**(4): 754–768.

Naderi B and Azab A (2014). Modeling and heuristics for scheduling of distributed job shops. *Expert Systems with Applications* **41**(17): 7754–7763.

Ouelhadj D and Petrovic S (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling* **12**(4): 417–431.

Pan QK and Ruiz R (2012). An estimation of distribution algorithm for lot-streaming flowshop problems with setup times. *Omega* **40**(2): 166–180.

Qin H, Zhang Z, Qi Z and Lim A (2014). The freight consolidation and containerization problem. *European Journal of Operational Research* **234**(1): 37–48.

Shen JN, Wang L and Wang SY (2015). A bi-population EDA for solving the no-idle permutation flow-shop scheduling problem with the total tardiness criterion. *Knowledge-Based Systems* **74**: 167–175, http://www.sciencedirect.com/science/article/pii/S095070511400416X.

Taguchi G (1986). *Introduction to Quality Engineering*. Asian Productivity Organization: Tokyo.

Tasgetiren MF, Liang YC, Sevkli M and Gencyilmaz G (2007). A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. *European Journal of Operational Research* **177**(3): 1930–1947.

Tzeng YR, Chen CL and Chen CL (2012). A hybrid EDA with ACS for solving permutation flowshop scheduling. *The International Journal of Advanced Manufacturing Technology* **60**(9–12): 1139–1147.

Valdez F, Melin P and Castillo O (2011). An improved evolutionary method with fuzzy logic for combining particle swarm optimization and genetic algorithms. *Applied Soft Computing* **11**(2): 2625–2632.

Valdez F, Melin P and Castillo O (2014). A survey on nature-inspired optimization algorithms with fuzzy logic for dynamic parameter adaptation. *Expert Systems with Applications* **41**(14): 6459–6466.

Wang K and Choi SH (2012). A decomposition-based approach to flexible flowshop scheduling under machine breakdown. *International Journal of Production Research* **50**(1): 215–234.

Wang K and Choi SH (2014). A holonic approach to flexible flowshop scheduling under stochastic processing times. *Computers & Operations Research* **43**: 157–168, http://www.sciencedirect.com/science/article/pii/S0305054813002839.

Wang L, Fang C, Suganthan PN and Liu M (2014). Solving system-level synthesis problem by a multi-objective estimation of distribution algorithm. *Expert Systems with Applications* **41**(5): 2496–2513.

Wang L, Wang SY and Xu Y (2012). An effective hybrid EDA-based algorithm for solving multidimensional knapsack problem. *Expert Systems with Applications* **39**(5): 5593–5599.

Wang K, Choi SH, Qin H and Huang Y (2013a). A cluster-based scheduling model using SPT and SA for dynamic hybrid flow shop problems. *The International Journal of Advanced Manufacturing Technology* **67**(9–12): 2243–2258.

Wang L, Wang S and Liu M (2013b). 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 SY, Wang L, Liu M and Xu Y (2013c). 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 SY, Wang L, Liu M and Xu Y (2013d). An enhanced estimation of distribution algorithm for solving hybrid flow-shop scheduling problem with identical parallel machines. *The International Journal of Advanced Manufacturing Technology* **68**(9–12): 2043–2056.

Xu Y, Wang L, Wang S and Liu M (2014). An effective hybrid immune algorithm for solving the distributed permutation flow-shop scheduling problem. *Engineering Optimization* **46**(9): 1269–1283.

Yalaoui N, Ouazene Y, Yalaoui F, Amodeo L and Mahdi H (2013). Fuzzy-metaheuristic methods to solve a hybrid flow shop scheduling problem with pre-assignment. *International Journal of Production Research* **51**(12): 3609–3624.

Zadeh LA (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man and Cybernetics*. SMC-3(1): 28–44.

Zandieh M and Gholami M (2009). An immune algorithm for scheduling a hybrid flowshop with sequence-dependent setup times and machines with random breakdowns. *International Journal of Production Research* **47**(24): 6999–7027.

Zhang Z, Che O, Cheang B, Lim A and Qin H (2013). A memetic algorithm for the multiperiod vehicle routing problem with profit. *European Journal of Operational Research* **229**(3): 573–584.

Zhang L, Gao L and Li X (2013). A hybrid genetic algorithm and tabu search for a multi-objective dynamic job shop scheduling problem. *International Journal of Production Research* **51**(12): 3516–3531.

Zhang Y and Li X (2011). Estimation of distribution algorithm for permutation flowshops with total flowtime minimization. *Computers & Industrial Engineering* **60**(4): 706–718.

Received 7 November 2014;
accepted 3 June 2015 after two revisions