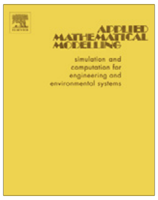
Applied Mathematical Modelling 37 (2013) 2737–2746

Contents lists available at SciVerse ScienceDirect

Applied Mathematical Modelling

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



Multiple parents crossover operators: A new approach removes the overlapping solutions for sequencing problems

Shih-Hsin Chen [a], Min-Chih Chen [b], Pei-Chann Chang [c,*], V. Mani [d]

[a] Department of Electronic Commerce Management, Nanhua University, No. 55, Sec. 1, Nanhua Rd., Zhongkeng, Dalin Township, Chiayi County 62248, Taiwan, ROC
[b] Department of Information Management, WuFeng University, Chiayi County 62153, Taiwan, ROC
[c] Department of Information Management, Yuan-Ze University, 135 Yuan-Dong Rd., Taoyuan 32026, Taiwan, ROC
[d] Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India

ARTICLE INFO

Article history:
Received 3 October 2010
Received in revised form 20 May 2012
Accepted 4 June 2012
Available online 15 June 2012

Keywords:
Diversity
Removing redundant solutions
Multi-parents crossover operator
Flowshop scheduling problems

ABSTRACT

Maintaining population diversity throughout generations of Genetic Algorithms (GAs) is key to avoid premature convergence. Redundant solutions is one cause for the decreasing population diversity. To prevent the negative effect of redundant solutions, we propose a framework that is based on the multi-parents crossover (MPX) operator embedded in GAs. Because MPX generates diversified chromosomes with good solution quality, when a pair of redundant solutions is found, we would generate a new offspring by using the MPX to replace the redundant chromosome. Three schemes of MPX will be examined and will be compared against some algorithms in literature when we solve the permutation flowshop scheduling problems, which is a strong NP-Hard sequencing problem. The results indicate that our approach significantly improves the solution quality. This study is useful for researchers who are trying to avoid premature convergence of evolutionary algorithms by solving the sequencing problems.

1. Introduction

Genetic Algorithms (GAs) have been widely used to solve many optimization problems because of their ease of use with promising results. GAs maintain and evolve solutions through the selection and variation in each generation. Through the generations of GAs, the population converges to better solution space while the population diversity is decreased in the same time. Premature convergency is well-recognized for GAs [1–3], which causes the problem of staying at local optimal instead of the global optimum. It is because that evolutionary algorithms attempt to converge to a optimal solution so that the solution space is narrowed down to a small region. Hence, it is a key to improve the population diversity when the population diversity is poor. As a result, many researchers studied some approaches to avoid the premature convergence of GAs, including:

1. Restart Strategy [4,5].
2. Immigrants [6–10].
3. Multiple crossover operators [11–16].
4. Adaptive Strategy [17–19].

---

* Corresponding author.

E-mail addresses: shihhsin@mail.nhu.edu.tw (S.-H. Chen), cthunter@mail.wfc.edu.tw (M.-C. Chen), iepchang@saturn.yzu.edu.tw (P.-C. Chang), mani@aero.iisc.ernet.in (V. Mani).

0307-904X/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.apm.2012.06.005

No matter what kind of strategy is used, Jin and Branke [20] pointed out the maintenance population diversity throughout the run is able to deal with the premature convergence. When the premature convergence is occurred, redundant solutions (or called overlapping solutions) are the one of reasons which causes the poor population diversity [21,22]. Redundant solutions usually occur in the evolutionary progress. Population diversity is decreased rapidly when redundant solutions are accumulated continually in the population. Because of this problem, [21,22] suggested that overlapping solutions should be removed in each run. It is proved that the removal of redundant solutions improves the performance. Other researches [23–25] eliminate the similar parents by using the lowest fitness chromosome in the group. Although the replacement strategies can promote population diversity, the disadvantage is the lowest fitness chromosome may not be selected in the next run.

The aim of this paper is to propose a new framework which enhances the population diversity by using the removal of the redundant solutions when we want to solve the NP-Hard sequencing problems. The major approaches in this framework are to detect the redundant solutions first and then utilize a multi-parents crossover (MPX) operator to generate new chromosomes to replace the overlapping solutions. MPX stemmed from [26] who shown it is possible to use more than two parents in crossover operation. A MPX operator can generate a more diversified population than two parents in sequencing problems [27,28]. Besides the diversity concern, it is showed that MPX can lead the GAs to better performance [29,15] because the MPX remain takes the advantage of the sequencing information in the population. Consequently, the new offsprings generated by MPX maintain the population diversity without degrading the solution quality too much.

The rest of the paper is organized as follows: Section 2 describes the formulations of PFSPs. Section 3 shows the methodology to generate diversified and good fitness solutions by using the MPX. Section 4 shows experiment results and discussion. Section 5 is the conclusions of this research.

2. Problem statements

Flowshops provide a convenient means to model serial manufacturing processes. The flowshop is a processing facility that consists of several machines on which jobs are processed in a sequential manner. In the permutation flowshop problem (PFSP), all the jobs follow the same processing order on each of the machines. The PFSP to minimize the makespan can be defined as follows:

Suppose there are n jobs and m machines. Let $p(i,j),\ 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m$, be the processing time of job i on machine j and $\pi = (\pi_1, \ldots, \pi_n)$ be a job permutation (i.e., processing order of the jobs). Then the completion times $C(\pi_i, j)$ are calculated as follows:

$$C(\pi_1, 1) = p(\pi_1, 1), \tag{1}$$

$$C(\pi_i, 1) = C(\pi_{i-1}, 1) + p(\pi_i, 1) \quad \text{for } i = 2, \ldots, n, \tag{2}$$

$$C(\pi_1, j) = C(\pi_1, j-1) + p(\pi_1, j) \quad \text{for } j = 2, \ldots, m, \tag{3}$$

$$C(\pi_i, j) = \max\{C(\pi_{i-1}, j), C(\pi_i, j-1)\} + p(\pi_i, j) \quad \text{for } i = 2, \ldots, n;\ j = 2, \ldots, m. \tag{4}$$

The makespan is

$$C_{\max}(\pi) = C(\pi_n, m). \tag{5}$$

The objective is to find a permutation π^* that minimizes $C_{\max}(\pi)$.

3. Methodology

The theme of this paper is to remove the effect of redundant solutions at each generation. In order to introduce diversified chromosomes to the population, we illustrate a framework that will eliminate redundant solutions in GA by means of replacing the redundant solutions through the application of multi-parents crossover operator. Section 3.1 will explain the detailed procedures of the proposed framework. Since several MPX methods exist, the following Section 3.2 will explain the process of generating diversified solutions by these MPX operators. Finally, an application will explain how different schemes can generate an offspring.

3.1. A framework of removing overlapping solutions in GAs

It is natural that the similar chromosomes or the number of redundant solutions will increase during the evolutionary progress that will enlarge the solution space or the objective space. As a result, various methods were proposed to replace the overlapping solutions [21,22]. When an overlapping solution is detected, the next step is to generate a new solution in replacement of the overlapping one. In this study, we employ a different approach from the past – by using the multi-parent crossover operator embedded in GAs, to generate new solutions in replacement of redundant chromosomes during the

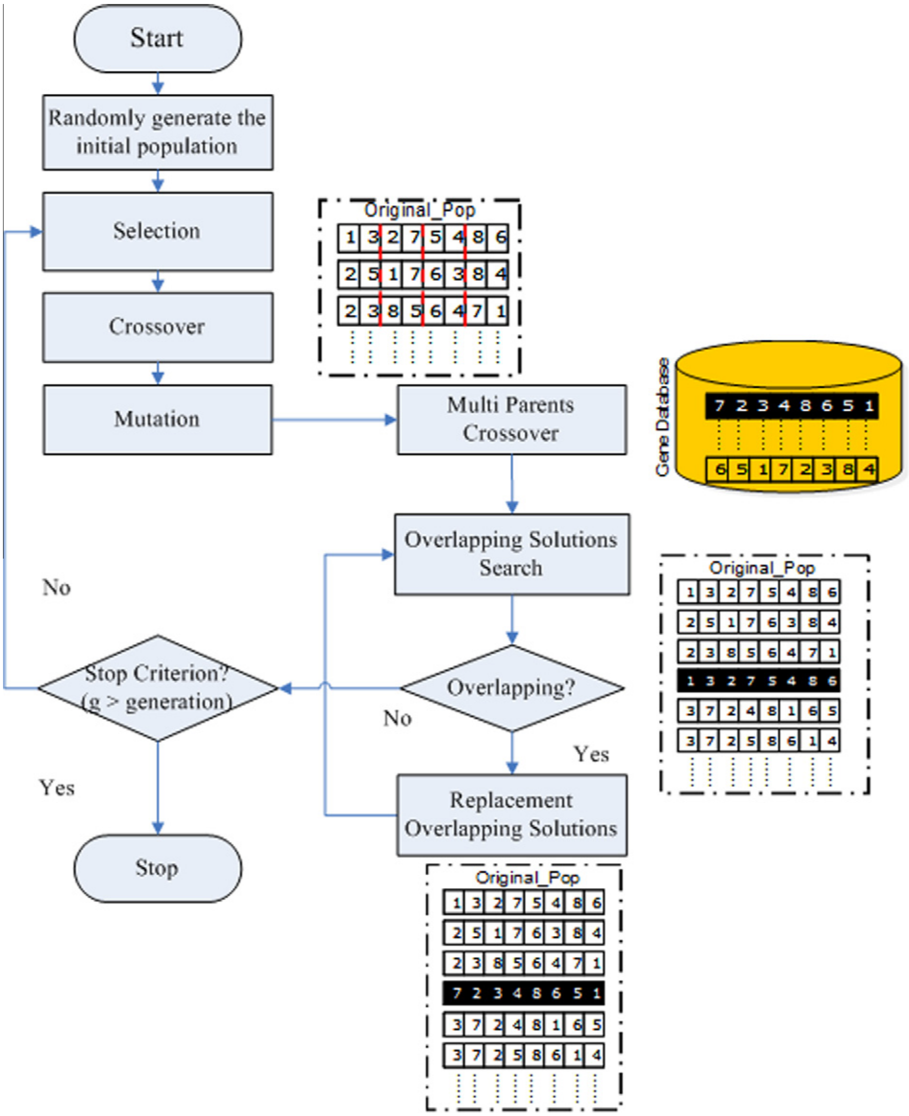


Fig. 1. A framework of removing overlapping solutions in GAs.

evolutionary progress. The detailed procedure is depicted in Fig. 1, and below is a detailed description of the procedures of the framework:

Step 1: Initialization
It is a common step of GAs. We initialize the population which consists a set of chromosomes. A chromosome represents a sequence for a problem we want to solve. The sequence is the processing sequence for the flowshop scheduling problem with integer numbers. Except that, the parameters of GAs are also initialized for the later on use.

Step 2: Fitness evaluation and selection operator
The chromosome fitness comes from the objective value of the sequence for the problem. Then, the selection operator choose better chromosomes to be survived. The binary tournament operator [30] is employed, which selects the better chromosomes with lower objective values in this minimization problem.

Step 3: Crossover operator
This study applies the two-point crossover operator to mate two chromosomes which are randomly selected. There is a crossover rate (P_c) which decides whether the crossover operator implements the mating of two chromosomes.

Step 4: Mutation operator

A chromosome is decided to mutated if it is less than the mutation rate (P_m). A famous swap mutation operator is used here. The approach is to generate two random cut-points of the chromosome, and then swap the genes belonged to the two cut-points.

Step 5: Multi-parents crossover operator

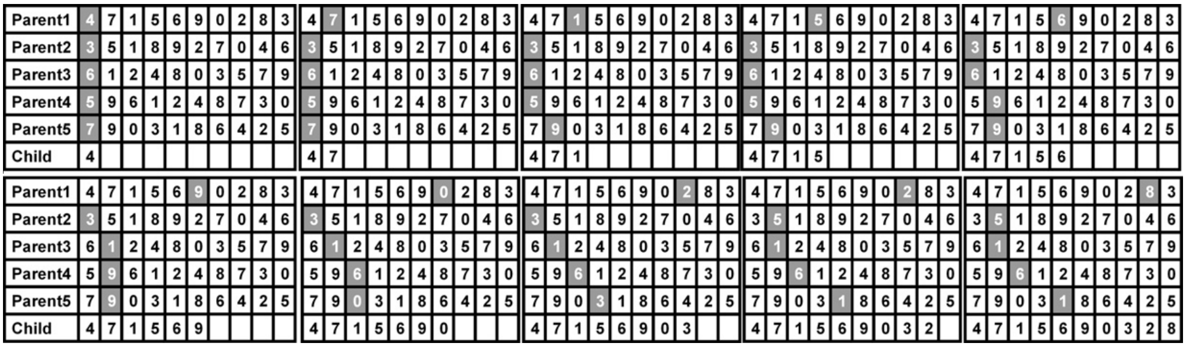
A multi-parent crossover operator is designed to generate offsprings from numerous parents. In this step, the generated offsprings are stored in an external archive and could be used in the replacement stage. The size of the external archive is equal to the population size. Then, all chromosomes are checked pair-wisely according to their own sequence results. When a redundant solution is found during the pair-wise comparison, we sequentially draw a chromosome from the external archive which has not been used and then replace the overlapping solution. Thus, it guarantees the simplicity of the proposed framework. About the way to generate new offsprings by the MPX operators, we introduce three approaches in the following subsection.

3.2. Three schemes of multi-parents crossover operator

Like the two-parents crossover in most GAs, there are numerous MPX operators for sequencing problems. For example, there are the gene reproduction mechanism and the multi-parent partially mapped crossover (MPPMX) proposed by Eiben et al. [29] and Ting et al. [15] respectively. In Eiben et al. [29], it presented the effect of increasing the number of parents from two to many. While it is observed that multi-parent crossovers can lead to better performance, the performance of those algorithms seems to be dependent on certain kinds of problems.

After an extensive review of MPX operator, we have come to the conclusion that two major characteristics of MPX operator are different from traditional two-parents crossover: scanning order and competition rule. Scanning order selects a gene from the candidate genes based on the order of each chromosome. Competition rule is a pre-determined rule by which that allows offsprings to inherit genes from the selection of multiple parents. Although there are many MPX operators, we pick up three schemes according to a distinct scanning order, which are Scanning Based Crossover (SBC), Adjacency Based Crossover (ABC), and Diagonal Based Crossover (DBC). We employ the same count of occurrence competition rule under these methods. The descriptions are as follows:

- SBC [29]: If the relation of genes is isolated in a chromosome, we can go through the sequence of the chromosome from the beginning to the end, and take the value of every parents as candidates. Then the occurrence competition rule is applied as selection criteria. The process is continued in the same manner until scanning to the last position. In order to present the scheme clearly, we demonstrate the above mentioned SBC in Fig. 2. Each sub-figure determines the selected gene from the beginning to the end.
- ABC [31]: If the relative position of genes in a chromosome or the sequence of composition in a gene fragment is important (e.g., some chromosomes may share the same arcs in the traveling salesman problems), ABC would be suitable in this case. We interpret the ABC in the following way: the first gene value in the first parent is inherited in the beginning and we use the gene value to update the marker; then separately, we search the adjacent gene value from all chromosomes according to the marker value; the second gene value is inherited from competition and updates the marker value; the process is continued in the same manner until all the assignments are done. In order to present the scheme clearly, we demonstrate the above mentioned ABC in Fig. 3.
- DBC [32]: In sequencing problems, it is quite likely that there are many genes blocks within chromosomes. Some researchers attempt to utilize this information in the mating progress. Diagonal Based Crossover (DBC) is a technique which divides the genes into blocks by the number of parents and then forms a complete chromosome by the diagonal rule. In order to present the scheme clearly, we demonstrate the above mentioned DBC in Fig. 4.



Parent1	4	7	1	5	6	9	0	2	8	3
Parent2	3	5	1	8	9	2	7	0	4	6
Parent3	6	1	2	4	8	0	3	5	7	9
Parent4	5	9	6	1	2	4	8	7	3	0
Parent5	7	9	0	3	1	8	6	4	2	5
Child	4									

Parent1	4	7	1	5	6	9	0	2	8	3
Parent2	3	5	1	8	9	2	7	0	4	6
Parent3	6	1	2	4	8	0	3	5	7	9
Parent4	5	9	6	1	2	4	8	7	3	0
Parent5	7	9	0	3	1	8	6	4	2	5
Child	4	7								

Parent1	4	7	1	5	6	9	0	2	8	3
Parent2	3	5	1	8	9	2	7	0	4	6
Parent3	6	1	2	4	8	0	3	5	7	9
Parent4	5	9	6	1	2	4	8	7	3	0
Parent5	7	9	0	3	1	8	6	4	2	5
Child	4	7	1							

Parent1	4	7	1	5	6	9	0	2	8	3
Parent2	3	5	1	8	9	2	7	0	4	6
Parent3	6	1	2	4	8	0	3	5	7	9
Parent4	5	9	6	1	2	4	8	7	3	0
Parent5	7	9	0	3	1	8	6	4	2	5
Child	4	7	1	5						

Parent1	4	7	1	5	6	9	0	2	8	3
Parent2	3	5	1	8	9	2	7	0	4	6
Parent3	6	1	2	4	8	0	3	5	7	9
Parent4	5	9	6	1	2	4	8	7	3	0
Parent5	7	9	0	3	1	8	6	4	2	5
Child	4	7	1	5	6					

Parent1	4	7	1	5	6	9	0	2	8	3
Parent2	3	5	1	8	9	2	7	0	4	6
Parent3	6	1	2	4	8	0	3	5	7	9
Parent4	5	9	6	1	2	4	8	7	3	0
Parent5	7	9	0	3	1	8	6	4	2	5
Child	4	7	1	5	6	9				

Parent1	4	7	1	5	6	9	0	2	8	3
Parent2	3	5	1	8	9	2	7	0	4	6
Parent3	6	1	2	4	8	0	3	5	7	9
Parent4	5	9	6	1	2	4	8	7	3	0
Parent5	7	9	0	3	1	8	6	4	2	5
Child	4	7	1	5	6	9	0			

Parent1	4	7	1	5	6	9	0	2	8	3
Parent2	3	5	1	8	9	2	7	0	4	6
Parent3	6	1	2	4	8	0	3	5	7	9
Parent4	5	9	6	1	2	4	8	7	3	0
Parent5	7	9	0	3	1	8	6	4	2	5
Child	4	7	1	5	6	9	0	3		

Parent1	4	7	1	5	6	9	0	2	8	3
Parent2	3	5	1	8	9	2	7	0	4	6
Parent3	6	1	2	4	8	0	3	5	7	9
Parent4	5	9	6	1	2	4	8	7	3	0
Parent5	7	9	0	3	1	8	6	4	2	5
Child	4	7	1	5	6	9	0	3	2	

Parent1	4	7	1	5	6	9	0	2	8	3
Parent2	3	5	1	8	9	2	7	0	4	6
Parent3	6	1	2	4	8	0	3	5	7	9
Parent4	5	9	6	1	2	4	8	7	3	0
Parent5	7	9	0	3	1	8	6	4	2	5
Child	4	7	1	5	6	9	0	3	2	8

Fig. 2. The Example of illustration of Scanning Based Crossover (SBC).

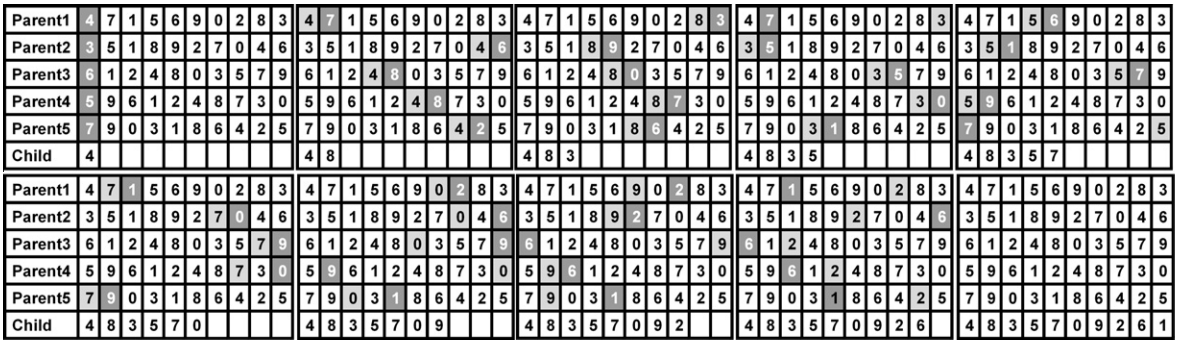


Fig. 3. The Example of illustration of Adjacency Based Crossover (ABC).

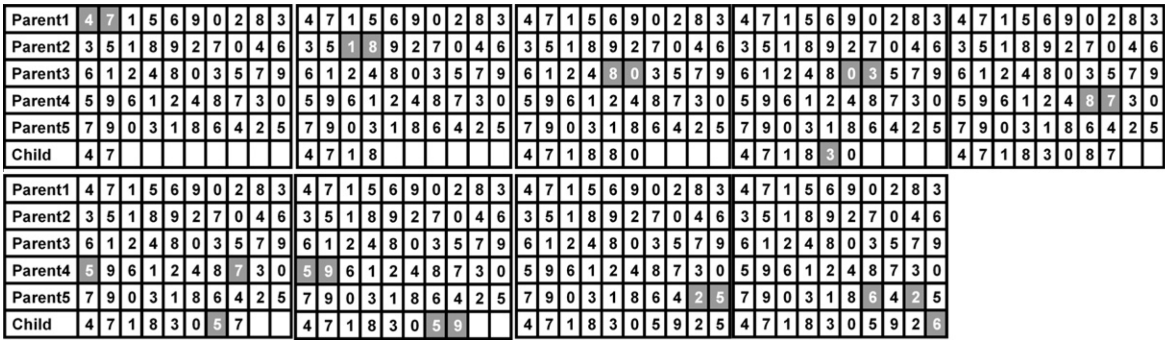


Fig. 4. The Example of illustration of Diagonal Based Crossover (DBC).

Finally, when these MPX operators work with the proposed framework, the three new algorithms are named as GA-RO$_{SBC}$, GA-RO$_{ABC}$, and GA-RO$_{DBC}$ where the RO is the abbreviation for removing overlapping solutions. According to the research of Eiben et al. [29], which MPX operator would lead to better solution quality in the NP-Hard flowshop scheduling problem is unknown, but through this research, we conducted extensive experiments to select a better MPX operator. In the next section, we will compare our approach with some existing algorithms.

4. Computational results

We conducted experiments to evaluate the performance of the proposed framework. There are three multi-parents crossover operators which is used to replace the overlapping solutions. The experiments were tested on the PFSPs to minimize the makespan problem. The testing 110 Taillard instances [33] are drawn from OR-Library[1]. There are four groups of job sizes: 20, 50, 100, and 200. In each job group, they have 5, 10, and 20 machines to process the jobs. The stopping criterion is to examine $500 * 2 * n$ solutions where n is the number of jobs. The parameters of GAs include the crossover rate, mutation rate, and population size which were determined in our preliminary experiments. They were set up as 0.6, 0.3, and 100, respectively. In addition, the number of parents is an important parameter because it influences the population diversity a lot. The setting of the number of parents is also different in different problems, this parameter is also decided by Design-of-Experiment (DOE). In this study, the number of parents was set up as 3 in later experiments.

To compare the proposed algorithm with others in literature, the average error ratio (Δ_{avg}) is used to evaluate the performance of all the algorithms. The error ratio of a solution X_i generated by an algorithm is calculated as follows:

$$\Delta_{avg} = \frac{\sum_{i=1}^{R} \frac{Heu_i - Best_i}{Best_i} \times 100}{R},$$

where Heu_i is the solution given by any of the R replications of the considered algorithms and $Best_i$ is the makespan value of the best known or optimal solution provided by Taillard [33]. In all the experiments, we replicated each instance 30 times.

In order to determine which algorithm is statistically significant, ANOVA (Analysis of Variance) and Duncan grouping test were employed to further distinguish the performance of the algorithms. ANOVA is a statistic method for testing the difference between/among the factors. Once there exists significant difference, it means the factor (input variable) will influence

---

[1] <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>.

the output (i.e., the objective values or the response) [34]. As shown in ANOVA table, Source indicated the input factors of the statistic model, DF is the degree of freedom, SS is sum of squares, F is the value of F-test, and Pr is the probability of the statistic significance [34].

Because the *P*-value of the factor Method is significant, we employ the Duncan grouping test to verify the pair-wise performance. Duncan Table is the Duncan grouping test for all the compared algorithms. In this table, Mean is the average value and *N* is the number of the observations. In the Duncan grouping test, if two algorithms share the same alphabet (i.e., they are in the same group), there is no significant difference between them. Otherwise they are significantly different [34].

4.1. The performance of MPX operators in the proposed framework

Three MPX operators SBC, ABC, and DBC are likely to be utilized in the proposed framework to replace overlapping solutions. We denote the three new algorithms as GA-RO$_{SBC}$, GA-RO$_{ABC}$, and GA-RO$_{DBC}$. In Table 1, the factor method is very significant, Duncan comparison is carried out in Table 2.

The Duncan comparison (in objective values) reveal that as soon as GA considers the removing overlapping solutions (GA-RO$_{SBC}$, GA-RO$_{ABC}$, and GA-RO$_{DBC}$), the performance is better than the GAs without replacing the overlapping solutions. It could be very important for GAs to remove the redundant solutions because redundant solutions degrade the solution. The reason might be the overlapping solutions decreasing the population diversity. We investigate this issue in the next subsection.

When we compare the three schemes, they are different to each other in terms of solution quality. The GA-RO$_{DBC}$ is the best method among the three MPX operator. The second group is GA-RO$_{ABC}$ and the last one is GA-RO$_{SBC}$. However, all the schemes outperform GAs without removing overlapping solutions.

Table 3 showed the statistical average error ratio (Δ_{avg}) of the results and the statistical average cpu time (t_{avg}) of the results obtained by GA-RO$_{DBC}$ and GA tested on the 110 test instances from Taillard. Although GA-RO$_{DBC}$ is better than GA in term of solution quality, the GAs appears more efficient in terms of CPU time because GA-RO$_{DBC}$ requires additional time to detect redundant solutions and to replace the identical solutions generated by MPX operator.

Because GA-RO$_{DBC}$ is the best framework when we apply different MPX operators, it is used to compare the algorithms in literature. The detail result is presented in the next section.

4.2. Comparison with the relative performance of GAs

In this section, GA-RO$_{DBC}$ is compared with Artificial Chromosome with Genetic Algorithms [35], Guided Memetic Algorithm [36], and Mining Gene Genetic Algorithms [37]. These algorithms are discussed below:

- Artificial Chromosome with Genetic Algorithms (ACGA) [35]: This is our previously developed approach that combines an probabilistic model with a standard genetic algorithm. By using the hybrid framework, both global and location information is used.
- Guided Memetic Algorithm (GMA) [36]: Guided Memetic Algorithm. The probabilistic model is used to assist a local search operator to reduce the computational overhead.
- Mining Gene Genetic Algorithms (MGGA) [37]: This algorithm is designed for treating machine scheduling problems. The linear assignment algorithm and a greedy heuristic are embedded in MCGA.

Table 1
ANOVA results on the objective values of the flowshop scheduling problem produced by different schemes.

Source	DF	SS	Mean square	*F* value	*P* value
Instances	109	1.53252E + 11	1405983543	1372051	<.0001
Method	3	581338.9917	193779.6639	189.1	<.0001
Instances*method	327	842601.9333	2576.764322	2.51	<.0001
Error	12760	13075568.57	1024.731079		
Corrected total	13199	1.53267E + 11			

Table 2
Duncan grouping on the objective values of the flowshop scheduling problem produced by different schemes.

Duncan grouping	Mean	*N*	Method
A	5025.7012	3300	GA
B	5020.2206	3300	GA-RO$_{SBC}$
C	5012.7876	3300	GA-RO$_{ABC}$
D	5008.4855	3300	GA-RO$_{DBC}$

Table 3
Performance evaluation of algorithms for makespan criterion.

n × m	GA						GA-RO$_{DBC}$					
	Δ_{min}	Δ_{avg}	Δ_{max}	t_{min}	t_{avg}	t_{max}	Δ_{min}	Δ_{avg}	Δ_{max}	t_{min}	t_{avg}	t_{max}
20×5	0.25	1.32	3.36	0.07	0.08	0.10	0.24	1.15	2.20	0.17	0.20	0.21
20×10	0.71	2.38	5.05	0.09	0.10	0.13	0.59	1.93	3.69	0.20	0.22	0.24
20×20	0.56	1.95	3.53	0.13	0.15	0.16	0.35	1.53	3.25	0.24	0.26	0.28
50×5	0.20	0.83	1.88	0.47	0.49	0.51	0.06	0.63	1.46	1.10	1.13	1.17
50×10	1.40	3.38	5.60	0.60	0.62	0.65	1.19	2.72	4.62	1.22	1.25	1.28
50×20	3.03	4.64	6.55	0.89	0.92	0.94	2.78	4.15	5.63	1.48	1.51	1.54
100×5	0.16	0.55	1.15	2.59	2.65	2.71	0.11	0.45	0.89	5.67	5.78	5.88
100×10	0.94	2.05	3.39	3.02	3.07	3.11	0.72	1.69	3.02	6.04	6.19	6.29
100×20	2.61	4.10	5.82	3.93	4.05	4.16	2.32	3.60	5.13	7.03	7.11	7.20
200×10	0.44	1.11	1.96	17.47	17.77	18.13	0.24	0.77	1.49	33.18	35.71	37.92
200×20	1.80	2.89	4.18	21.57	21.74	21.94	1.53	2.59	3.74	34.12	37.42	39.64
Average	1.10	2.29	3.86	4.62	4.69	4.78	0.92	1.93	3.19	8.22	8.80	9.24

Table 4
Parameter settings of the implemented algorithms.

Method	Settings
GA-RO$_{DBC}$	Crossover rate = 0.6 Mutation rate = 0.3 Number of parents = 3
ACGA	Starting generation = 0.7$*$ (total generations) Interval = 0.1$*$ (total generations) Crossover rate = 0.6 Mutation rate = 0.3 $\lambda = 0.5$
GMA	Crossover rate = 0.6 Mutation rate = 0.3 $\lambda = 0.5$
MGGA	Interval: 0.05$*$(total generations) Crossover rate = 0.9 Mutation rate = 0.5
Common	The elitist solutions=10% terminational criterion=500 × $m \times n$ Population size = 100

Table 5
ANOVA results on the objective values of PFSP produced by different algorithms.

Source	DF	SS	Mean square	*F* value	*P* value
Instances	109	1.54256E + 11	1415188569	1511088	<.0001
Method	3	1421018.101	473672.7005	505.77	<.0001
Instances*method	327	7547203.024	23080.13157	24.64	<.0001
Error	12760	11950201.37	936.5361573		
Corrected total	13199	1.54276E + 11			

For the performance evaluation purpose, the parameter settings of the algorithms are decided by DOE to analysis the efficiency of single factor and interaction with another parameters. There were given in Table 4.

The ANOVA results are presented in Table 5 where the factor method is significant. We thus continue to evaluate the differences of the algorithms. It is evident from Table 6 that four algorithms were significantly different and GA-RO$_{DBC}$ performed significantly better than others in term of solution quality. Except for the information of ANOVA and Duncan comparisons, we supply the statistic results of these algorithms to deal with the last 50 Taillard instances in Table 7. From Table 7, it clearly showed that GA-RO$_{DBC}$ outperformed other GA-based algorithms. Thus, we may conclude that the removing redundant solutions together with a MPX operator is beneficial for the GAs.

4.3. Discussion

We take one more step to understand why the removing redundant solution is effective. We examine the population diversity of proposed algorithm GA-RO$_{DBC}$ and GA. The diversity metric is shown in Eq. 7. The foundation of the diversity

Table 6
Duncan grouping on the objective values of PFSP produced by different algorithms.

Duncan grouping	Mean	*N*	Method
A	5035.0624	3300	GMA
B	5016.3655	3300	MGGA
C	5011.1815	3300	ACGA
D	5008.4855	3300	GA-RO$_{DBC}$

Table 7
Partial Taillard flowshop results.

Instance	ACGA	GMA	MGGA	GA			GA-RO$_{DBC}$		
	Δ_{avg}	Δ_{avg}	Δ_{avg}	Δ_{min}	Δ_{avg}	Δ_{max}	Δ_{min}	Δ_{avg}	Δ_{max}
ta061	0.25	0.20	0.04	0.04	0.30	0.62	0.00	0.09	0.62
ta062	0.34	0.32	0.29	0.19	0.33	0.83	0.17	0.39	0.80
ta063	0.77	0.73	0.78	0.58	0.84	1.45	0.35	0.84	1.18
ta064	0.32	0.30	0.44	0.10	0.32	0.52	0.00	0.28	0.52
ta065	0.47	0.23	0.12	0.10	0.58	1.89	0.06	0.43	1.16
ta066	0.26	0.18	0.19	0.04	0.27	0.84	0.00	0.24	0.51
ta067	0.77	0.73	1.09	0.25	0.90	1.51	0.25	0.71	1.20
ta068	0.52	0.50	0.20	0.00	0.60	1.53	0.24	0.56	0.74
ta069	0.58	0.57	0.57	0.00	0.68	1.28	−0.11	0.55	1.10
ta070	0.48	0.39	0.44	0.26	0.65	1.09	0.11	0.44	1.09
ta071	2.06	2.31	1.71	0.85	2.29	4.09	0.42	1.61	3.17
ta072	1.61	1.84	0.90	0.60	1.75	2.97	0.58	1.56	2.91
ta073	1.55	1.74	1.65	0.70	1.77	2.48	0.21	1.78	3.03
ta074	2.46	2.74	2.57	1.48	2.55	3.69	0.96	2.28	3.49
ta075	2.51	2.95	1.80	1.84	3.25	4.44	1.55	2.88	4.68
ta076	1.52	1.68	0.76	0.34	1.68	2.47	0.38	1.15	2.98
ta077	1.92	2.01	2.07	1.25	2.25	3.27	1.11	1.96	2.70
ta078	1.48	1.87	1.39	1.15	2.07	3.87	0.96	1.56	2.73
ta079	1.37	1.52	1.26	0.83	1.52	3.09	0.70	1.36	2.51
ta080	1.03	0.97	0.93	0.36	1.37	3.57	0.36	0.81	1.98
ta081	3.46	4.53	3.09	2.44	3.66	5.31	2.62	3.65	5.00
ta082	3.56	4.55	2.67	1.85	3.83	5.74	1.80	3.35	5.39
ta083	3.59	4.56	3.88	2.41	3.92	5.95	2.15	3.42	4.57
ta084	3.72	4.65	2.27	2.62	4.10	5.99	2.81	3.69	5.15
ta085	3.94	4.88	3.32	2.83	4.24	6.43	2.90	3.93	5.20
ta086	3.30	4.35	2.67	2.39	4.07	5.61	2.14	3.56	5.60
ta087	4.35	5.18	2.74	3.46	4.72	6.24	2.50	3.89	6.10
ta088	4.44	5.16	3.74	3.18	4.57	6.23	2.29	4.11	5.79
ta089	3.90	4.95	3.75	2.65	4.35	5.93	2.32	3.75	4.53
ta090	3.37	4.00	3.38	2.32	3.56	4.78	1.70	2.67	3.99
ta091	0.50	0.64	0.20	0.22	0.63	1.46	0.21	0.55	1.39
ta092	0.91	1.31	1.21	0.60	1.10	1.71	0.28	0.81	1.64
ta093	0.63	0.75	1.44	0.37	0.99	1.61	0.20	0.60	1.06
ta094	0.44	0.43	0.66	0.03	0.46	0.77	−0.07	0.51	1.11
ta095	1.45	1.68	1.28	0.61	1.60	2.91	0.26	0.85	1.40
ta096	1.04	1.51	1.12	0.31	1.22	2.64	0.20	0.71	1.83
ta097	1.11	1.33	0.87	0.86	1.43	2.50	0.74	1.18	1.76
ta098	0.73	0.86	0.94	0.19	0.97	1.69	0.06	0.63	1.58
ta099	1.10	1.37	0.78	0.79	1.41	2.31	0.24	0.84	1.46
ta100	0.93	1.13	0.67	0.46	1.27	1.99	0.28	1.01	1.69
ta101	2.41	3.48	3.66	1.65	2.71	4.08	1.47	2.34	3.14
ta102	2.73	3.77	4.03	2.12	2.96	3.67	1.84	2.71	3.76
ta103	3.20	4.06	4.39	2.18	3.16	4.53	1.79	2.97	4.00
ta104	2.28	3.08	3.55	1.05	2.60	4.29	1.23	2.21	3.93
ta105	2.68	3.48	3.74	1.77	2.88	4.69	1.63	2.69	3.75
ta106	3.10	4.20	4.52	1.89	3.34	4.39	1.48	2.98	4.06
ta107	2.72	3.40	3.82	1.64	2.55	3.84	1.58	2.37	3.63
ta108	2.74	3.30	3.79	1.82	2.89	4.12	1.34	2.57	3.25
ta109	2.69	3.57	3.92	1.61	2.83	4.30	1.31	2.33	4.04
ta110	2.93	3.75	4.10	2.26	2.94	3.93	1.67	2.68	3.81
Average	1.92	2.35	1.99	1.19	2.14	3.30	0.98	1.82	2.85

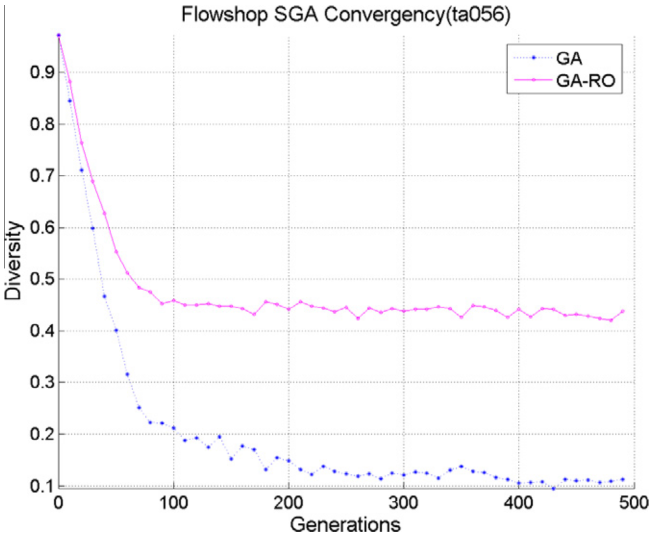


Fig. 5. Diversity analysis of the algorithms at instance ta056.

metric is based on the total hamming distance which sums the total distance among all solutions. Because we want to let the diversity value from 0 to 1, the total hamming distance is divided by the maximum estimated difference shown in Eq. (6). The parameter x and y represent the two chromosomes in a population P. The parameter k is chromosome length, and N is the size of population P. The $(k+1)*0.4*k$ estimates the max difference of a solution. The complete comparisons is $N*(N+1)/2$, here $N^2/2$ is the simplified meaning to represent it. Then, the maximum estimated difference is $(k+1)*0.4*k*N^2/2$. The diversity measure is obtained in Eq. (7).

$$Ham(x,y) = \sum_i |sgn(x[i] - y[i])|, \tag{6}$$

$$Diversity = \left[\sum_{i=1}^{N}\sum_{j=i+1}^{N} Ham(P[i],P[j])\right] \Big/ [(k+1)*0.4*k*N^2/2]. \tag{7}$$

We draw the instance ta056 (it is a scheduling problem of 50 jobs and 20 machines) and each algorithm is replicated 30 times. Fig. 5 depicts the diversity diagram against the generation of the compared algorithms. When GA does not eliminate the overlapping solutions, the diversity is decreased rapidly and the population diversity is poor after the generation 100. Because GA-RO$_{DBC}$ maintains good population diversity in a certain level, the algorithm has better chance to improve the solution quality. It might be the reason why GA-RO$_{DBC}$ is better than other GA-based algorithms.

5. Conclusion

The multi-parents crossover operator is one of the approaches that promotes population diversity in evolution algorithms. In this research, the multi-parent crossover was not substituted for the two-parent crossover operator. Instead, an entirely new framework is proposed for solving the overlapping solutions issue which degrades population diversity to a great extent. Under our approach, when a redundant solution is found, a new solution generated by MPX operator replaces the redundant solutions. We have proven that the removal of redundant solutions is significant in increasing diversity and enhancing algorithm performance. And with this method, our research presents an interesting framework to improve population diversity.

First, to better describe the multi-parents crossover operator, the MPX operator is applied to produce brand-new and promising chromosomes to replace redundant individuals in GAs. This method takes advantage of useful sequencing information in the population. MPX operator is able to generate diversified species without losing too much solution quality. Secondly, several multi-parents crossover operators are available to be used in the proposed framework. Experiments indicate that no matter what kinds of MPX operator is being used in the framework, the framework outperforms the GAs significantly. That said, this could be a good method for other evolutionary algorithms to improve their own performances. Finally, the benefit of removing redundant solutions is that while it provides sufficient diversity, it does not create an excessive diversity effect. We know if the population diversity is too high, GAs cannot be converged. Thus, the proposed framework provides moderate diversity during the evolutionary progress.

For the future development of multi-parent crossover embedded in frameworks for removing redundant solutions, our proposed framework is not limited to PFSPs. In addition, although the Diagonal Based Crossover operator may lead to better solution quality when used to replace overlapping solutions, this method does not guarantee better performance in other sequencing problems. We need to validate different MPX operators when we solve for new sequencing problems. Finally, due to the nature of MPX operators, the framework could be extended to solve other problems, such as multi-objective problems, uncertain environments, and dynamic optimization problems. Moreover, these schemes can be easily embedded in other advanced GAs, such as ACGA [35], Self-Guided GA [36,38], dominance properties with GAs [39], and Hybrid Heuristic method with GA [40]. MPX could be integrated with the above mentioned algorithms. Therefore, depending on the nature of the issues, we expect MPX operator to be used in different fields to introduce diversified solutions.

References

[1] N.F. McPhee, N.J. Hopper, Analysis of genetic diversity through population history, in: Proceedings of the Genetic and Evolutionary Computation Conference, vol. 2, 1999, Citeseer, pp. 1112–1120.
[2] E.K. Burke, S. Gustafson, G. Kendall, Diversity in genetic programming: an analysis of measures and correlation with fitness, IEEE Trans. Evolutionary Comput. 8 (1) (2004) 47–62.
[3] R. Burke, S. Gustafson, G. Kendall, A survey and analysis of diversity measures in genetic programming, in: Proceedings of the Genetic and Evolutionary Computation Conference Table of Contents, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002, pp. 716–723.
[4] W. Lin, T. Chen, Analysis of two restart algorithms, Neurocomputing 69 (16–18) (2006) 2301–2308.
[5] N. Raman, F. Talbot, The job shop tardiness problem: a decomposition approach, Eur. J. Operat. Res. 69 (2) (1993) 187–199.
[6] R. Ahuja, J. Orlin, A. Tiwari, A greedy genetic algorithm for the quadratic assignment problem, Comput. Operat. Res. 27 (10) (2000) 917–934.
[7] H. Cheng, S. Yang, Genetic algorithms with immigrants schemes for dynamic multicast problems in mobile ad hoc networks, Eng. Appl. Artificial Intell. 23 (5) (2010) 806–819.
[8] L. Xing, Y. Chen, K. Yang, F. Hou, X. Shen, H. Cai, A hybrid approach combining an improved genetic algorithm and optimization strategies for the asymmetric traveling salesman problem, Eng. Appl. Artificial Intell. 21 (8) (2008) 1370–1380.
[9] X. Yu, K. Tang, X. Yao, An immigrants scheme based on environmental information for genetic algorithms in changing environments, in: IEEE Congress on Evolutionary Computation, CEC 2008. (IEEE World Congress on Computational Intelligence), 2008, pp. 1141–1147.
[10] C. Azevedo, A. Araujo, Generalized immigration schemes for dynamic evolutionary multiobjective optimization, in: 2011 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2011, pp. 2033–2040.
[11] Z. Ahmed, Multi-parent extension of sequential constructive crossover for the travelling salesman problem, Int. J. Operat. Res. 11 (3) (2011) 331–342.
[12] C. António, A study on synergy of multiple crossover operators in a hierarchical genetic algorithm applied to structural optimisation, Struct. Multidiscip. Optim. 38 (2) (2009) 117–135.
[13] W.D. Chang, A multi-crossover genetic approach to multivariable PID controllers tuning, Expert Syst. Appl. 33 (3) (2007) 620–626.
[14] S. Esquivel, H. Leiva, R. Gallard, Multiple crossovers between multiple parents to improve search inevolutionary algorithms, in: Proceedings of the 1999 Congress on Evolutionary Computation, CEC 99, vol. 2, 1999.
[15] C.K. Ting, C.H. Su, C.N. Lee, Multi-parent extension of partially mapped crossover for combinatorial optimization problems, Expert Syst. Appl. 37 (3) (2010) 1879–1886.
[16] H.S. Yoon, B.R. Moon, An empirical study on the synergy of multiple crossover operators, IEEE Trans. Evolution. Comput. 6 (2) (2002) 212–223.
[17] A.G. Carvalho, A.F.R. Araujo, Improving NSGA-II with an adaptive mutation operator, in: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference, ACM, 2009, pp. 2697–2700.
[18] A. Eiben, Z. Michalewicz, M. Schoenauer, J. Smith, Parameter Control in Evolutionary Algorithms, Springer, 2007.
[19] S. Smit, A. Eiben, Parameter tuning of evolutionary algorithms: generalist vs specialist, Appl. Evolution. Comput. (2010) 542–551.
[20] Y. Jin, J. Branke, Evolutionary optimization in uncertain environments-a survey, IEEE Trans. Evolution. Comput. 9 (3) (2005) 303–317.
[21] Y. Nojima, K. Narukawa, S. Kaige, H. Ishibuchi, Effects of removing overlapping solutions on the performance of the NSGA-II algorithm, in: Proceedings of Third International Conference on Evolutionary Multi-criterion Optimization, Springer, 2005, pp. 341–354. vol. 3410.
[22] M. Sato, H. Aguirre, K. Tanaka, Effects of δ-Similar Elimination and Controlled Elitism in the NSGA-II Multiobjective Evolutionary Algorithm, in: Proceedings of the Congress on Evolutionary Computation, 2006, pp. 3980–3987.
[23] W. Cedeño, The multi-niche crowding genetic algorithm: analysis and applications, Ph.D. Thesis, University of California, 1995.
[24] W. Cedeño, V. Vemuri, Analysis of speciation and niching in the multi-niche crowding GA, Theor. Comput. Sci. 229 (1) (1999) 177.
[25] M. Lozano, F. Herrera, J. Cano, Replacement strategies to preserve useful diversity in steady-state genetic algorithms, Inform. Sci. 178 (23) (2008) 4421–4433.
[26] H. Muhlenbein, Parallel genetic algorithms in combinatorial optimization, in: O. Balci, R. Sharda, S. Zenios (Eds.), Computer Science and Operations Research, Pergamon Press, New York, 1992, pp. 441–456.
[27] D. Porumbel, J. Hao, P. Kuntz, Diversity control and multi-parent recombination for evolutionary graph coloring algorithms, in: Proceedings of the 9th European Conference on Evolutionary Computation in Combinatorial Optimization, Springer, 2009, pp. 121–132.
[28] S. Tsutsui, M. Yamamura, T. Higuchi, Multi-parent recombination with simplex crossover in real coded genetic algorithms, in: Proc. of the Genetic and Evolutionary Computation Conference (GECCO-99), 1999, pp. 657–664.
[29] A. Eiben, P. Raué, Z. Ruttkay, Genetic algorithms with multi-parent recombination, Lecture Notes Comput. Sci. 866 (1994) 78–87.
[30] D.E. Goldberg, et al., Genetic Algorithms in Search, vol. 432, 1989.
[31] A. Eiben, C. Van Kemenade, J. Kok, Orgy in the computer: multi-parent reproduction in genetic algorithms, Lecture Notes Comput. Sci. 929 (1995) 934–945.
[32] J. Lis, A. Eiben, A multi-sexual genetic algorithm for multiobjective optimization, in: IEEE International Conference on Evolutionary Computation, 1997, pp. 59–64.
[33] E. Taillard, Benchmarks for basic scheduling instances, Eur. J. Operat. Res. 64 (1993) 278–285.
[34] D. Montgomery, Design and Analysis of Experiments, John Wiley & Sons Inc, 2008.
[35] P.C. Chang, S.H. Chen, C.Y. Fan, Genetic algorithm integrated with artificial chromosomes for multi-objective flowshop scheduling problems, Appl. Math. Comput. 205 (2) (2008) 550–561.
[36] S. Chen, P. Chang, Q. Zhang, C. Wang, A Guided Memetic Algorithm with Probabilistic Models, Int. J. Innovative Comput. Inform. Control 5 (12B) (2009) 4753–4764.
[37] P.C. Chang, S.H. Chen, C.Y. Fan, Mining gene structures to inject artificial chromosomes for genetic algorithm in single machine scheduling problems, Appl. Soft Comput. J. 8 (1) (2008) 767–777.
[38] S.H. Chen, P.C. Chang, T.C.E. Cheng, Q. Zhang, A self-guided genetic algorithm for permutation flowshop scheduling problems, Comput. Operat. Res. 39 (7) (2012) 1450–1457.
[39] P.C. Chang, S.H. Chen, V. Mani, A hybrid genetic algorithm with dominance properties for single machine scheduling with dependent penalties, Appl. Math. Model. 33 (1) (2009) 579–596.
[40] D.Z. Zheng, L. Wang, An effective hybrid heuristic for flow shop scheduling, Int. J. Adv. Manufac. Technol. 21 (1) (2003) 38–44.