

## Heuristic based genetic algorithms for the re-entrant total completion time flowshop scheduling with learning consideration

Jianyou Xu<sup>a</sup>, Win-Chin Lin<sup>b</sup>, Junjie Wu<sup>c</sup>, Shuenn-Ren Cheng<sup>d</sup>  
Zi-Ling Wang<sup>b</sup>, and Chin-Chia Wu<sup>b\*</sup>

<sup>a</sup> College of Information Science and Engineering,  
Northeastern University, Shenyang 110819, China,  
xujianyou@mail.neu.edu.cn

<sup>b</sup> Department of Statistics, Feng Chia University,  
Taichung, Taiwan  
linwc@fcu.edu.tw; iloveeat7839@gmail.com; cchwu@fcu.edu.tw

<sup>c</sup> Management School, Zhejiang Shuren University,  
Zhejiang, China  
Wjj58@163.com

<sup>d</sup> Graduate Institute of Business Administration,  
Cheng Shiu University, Kaohsiung, Taiwan  
k0252@gcloud.csu.edu.tw

Received 14 February 2016

Accepted 18 July 2016

### Abstract

Recently, both the learning effect scheduling and re-entrant scheduling have received more attention separately in research community. However, the learning effect concept has not been introduced into re-entrant scheduling in the environment setting. To fill this research gap, we investigate re-entrant permutation flowshop scheduling with a position-based learning effect to minimize the total completion time. Because the same problem without learning or re-entrant has been proved NP-hard, we thus develop some heuristics and a genetic algorithm (GA) to search for approximate solutions. To solve this problem, we first adopt four existed heuristics for the problem; we then apply the same four methods combined with three local searches to solve the proposed problem; in the last stage we develop a heuristic-based genetic algorithm seeded with four good different initials obtained from the second stage for finding a good quality of solutions. Finally, we conduct experimental tests to evaluate the behaviours of all the proposed algorithms when the number of re-entrant times or machine number or learning effect or job size changes.

*Keywords:* re-entrant flowshop; learning effect; heuristic-based genetic algorithm.

### 1. Introduction

Most of classical scheduling models consider that the job processing time is fixed and constant<sup>1,2,3</sup>. However, it is invalid in many practical situations. For example, a steady decline in processing times usually takes place by performing the same task repeatedly<sup>4</sup>. For another example, each basic operation used in the manufacture of a product has its own learning function<sup>5</sup>. In such a

situation, the processing time of a job may take into consideration a constant occurrence and dynamic feature, or called as “the learning effect”.

The importance of learning effect has been presented in many practical situations. Applications of the learning effect can be found in machine shop<sup>5</sup>, in different branches of industry and for a variety of corporate activities<sup>6</sup>, and in shortening life cycles and an increasing diversity of products the manufacturing

environment<sup>7</sup>. However, the notion of learning is not introduced to the field of scheduling until Biskup<sup>4</sup>, and Cheng and Wang<sup>8</sup>. Since then, scheduling with learning considerations has received more attention in the scheduling research community for as long as 15 years. For details of research results on variants of the scheduling problem with different learning settings, the reader may refer to the research study by Biskup<sup>9</sup>.

The existing literature indicates that most scheduling with learning considerations considers single-machine case<sup>10-16</sup>, but the concept of learning process on scheduling flowshop are relatively limited. For example, Wang and Xia<sup>17</sup> provided the worst-case bound of the shortest processing time first (SPT) rule for the makespan and the total completion time problems for the flowshop scheduling problem with a learning effect. Wu and Lee<sup>18</sup> considered a permutation flowshop scheduling problem with a learning effect to minimize the sum of completion time. They proposed a dominance rule and several lower bounds used in the branch-and-bound algorithm to solve the problem. Wang and Wang<sup>19</sup> considered the processing time of a job as an exponent function of its position in a processing permutation and discussed the worst cases for the four regular performance criteria, namely, the total completion time, the total weighted completion time, the discounted total weighted completion time, and the sum of the quadratic job completion times for the flowshop scheduling problems. Chung and Tong<sup>20</sup> discussed a bi-criteria scheduling problem in an  $m$ -machine permutation flowshop environment with varied learning effects on different machines where the objective is to minimize the weighted sum of the total completion time and the makespan. They proposed a dominance criterion and a lower bound to accelerate the branch-and-bound algorithm for deriving the optimal solution. Kuo et al.<sup>21</sup> assumed that the time-dependent learning effect of a job was a function of the total normal processing time of jobs scheduled before the job and investigated worst-case analysis for the objective functions such as the makespan, the total flowtime, the sum of weighted completion times, the sum of the  $k$ th power of completion times, and the maximum lateness on the flowshop setting. Cheng et al.<sup>22</sup> studied a two-machine flowshop scheduling problem with a truncated learning function where they considered the actual processing time of a job as a function of the job's position in a schedule and the learning truncation parameter. The objective is to minimize the makespan. They proposed a branch-and-bound and three crossover-based genetic algorithms (GAs) to find the optimal and approximate solutions for the problem. Cheng<sup>23</sup>

considered a permutation flowshop scheduling problem with a position-dependent exponential learning effect to minimize the performance criteria of makespan and the total flow time. They showed Johnson's rule is not an optimal algorithm for the two-machine flow shop case and used the shortest total processing times first (STPT) rule to construct the worst-case performance ratios for the problems. More recently, Sun et al.<sup>24</sup> considered flow shop scheduling problems with a function of its position in a processing permutation. They provided some heuristic algorithms by using the optimal permutations for the corresponding single machine scheduling problems and discussed the worst-case bound of these heuristics. Sun et al.<sup>25</sup> studied some permutation flow shop scheduling problems with a general non-increasing function of its scheduled position, and presented approximation algorithms based on the optimal permutations for the corresponding single machine scheduling problems and analyze their worst-case error bound. For more papers with learning considerations, we suggest reader refer to the research studies by Janiak *et al.*<sup>26</sup>.

Generally, literature researchers often assume that a job is processed at most once by a machine on scheduling with learning considerations. However, this assumption does not hold in some real-life situations. For example, a job may be processed by the same machine more than once in semiconductor wafer manufacturing<sup>27</sup>, in signal processing<sup>28</sup>, and in printed circuit board manufacturing<sup>29-32</sup>. Such a setting is known as the "re-entrant flowshop". For details of research results on variants of the re-entrant scheduling problem, we refer the reader to three well-known survey papers<sup>32-34</sup>.

So far, there is only one research study on scheduling with the learning effect in the re-entrant flowshop setting. Wu *et al.*<sup>35</sup> considered the re-entrant permutation flowshop scheduling problem with a sum-of-processing-times-based learning function to minimize the makespan. They developed four heuristics by combining Johnson's rule with four local search methods that are effective in treating the flowshop scheduling problem and a simulated annealing (SA) algorithm to find near-optimal solutions for the problem. In view of these observations, in this paper we study the re-entrant permutation flowshop scheduling problem with a position-based learning function to minimize the total completion time which is a criterion popularly considered in the field of scheduling.

The organization of this paper is as follows: In Section 2 we define some notation and describe the problem. In Section 3 we build fourteen heuristics by

combining four well-known rules with three local search methods that are confirmed well in solving the flowshop scheduling problem without learning considerations and a genetic algorithm (GA) to search for approximate solutions. In Section 4 we conduct extensive experimental results which are included some small job-size and big-size jobs to test the performance of the proposed algorithms when changing the number of re-entrant times, machine number, and learning effect. Finally, we provide some conclusions and future suggestions in the last section.

### 2. Problem definition and description

Consider an  $m$ -machine flowshop in which there are  $n$  jobs to be processed on the  $m$  machines (i.e.,  $M_1, M_2, \dots, M_m$ ) in the same order and the job sequence is the same on all the machines. Consider that  $n$  jobs are ready at time zero for all the machines throughout the scheduling period and no preemption is allowed. Suppose that the waiting space between the machines is unlimited for the  $n$  jobs to be processed. In the re-entrant permutation flowshop setting, each job visits on the  $m$  machines following the same order, that is  $M_1^1, M_2^1, \dots, M_m^1; M_1^2, M_2^2, \dots, M_m^2; \dots$ ; and  $M_1^R, M_2^R, \dots, M_m^R$ , with up to  $R$  re-entrant times, starting on machine  $M_1$  and ending on  $M_m$ , and the job sequence is the same on any machine at each level. For clarification, we assume that each level has its own position-based learning effect, i.e., the actual processing time of  $J_{ji}^l$  to be scheduled in the  $k$ th position of a given schedule at level  $l$  on machine  $i$  is given by

$$P_{[k]i}^l = P_{ji}^l k^\alpha,$$

where  $P_{ji}^l$  denotes the processing time of  $J_{ji}^l$  on machine  $i$  at level  $l$ ,  $[k]$  denotes the job scheduled in the  $k$ th position of a sequence,  $i = 1, 2, \dots, m; k = 1, 2, \dots, n; l = 1, 2, \dots, R$ , and  $\alpha < 0$  is the learning index. The objective of this paper is to minimize the total completion time.

Let  $C_{[k]i}^l$  denote the completion time of  $J_{ji}^l$  scheduled in the  $k$ th position of a sequence on machine  $i$  at level  $l$ . Therefore,

$$C_{[k]i}^l = \max(C_{[k-1]i}^l, C_{[k-1]j}^l) + P_{[k]i}^l k^\alpha,$$

and the total completion time is  $\sum_{k=1}^n C_{1[k]m}^R$ .

To illustrate the working of the above definition, we provide an example in the following.

**Example 1.** Table 1 shows the data of a problem instance with  $n = 3$  jobs,  $m = 2$  machines, and  $R = 2$  re-entrant times, and the learning index is  $a = -0.152$ .

**Table 1.** The illustrative data

$l = 2$	$J_1$	$J_2$	$J_3$
$M_1^1$	10	8	7
$M_2^1$	5	9	6
$M_1^2$	3	7	5
$M_2^2$	1	4	6

Consider a job sequence  $(J_1, J_2, J_3)$  as example. The completion times of three jobs on machine  $M_2$  at level one are given as follows.

$$C_{[1]2}^1 = C_{[1]1}^1 + P_{[1]2}^1 = 10 + 5 = 15$$

$$C_{[2]2}^1 = \max(C_{[2]1}^1, C_{[1]2}^1) + P_{[2]2}^1 = \max(10 + 8 \times 2^{-0.152}, 15) + 9 \times 2^{-0.152} = 25.3$$

$$C_{[3]2}^1 = \max(C_{[3]1}^1, C_{[2]2}^1) + P_{[3]2}^1 = 25.3 + 6 \times 3^{-0.152} = 30.4$$

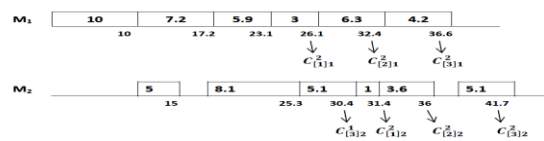
In what follows, the completion times of three jobs on machine  $M_2$  at level two are given as follows.

$$C_{[1]2}^2 = \max(C_{[1]1}^2, C_{[3]1}^1) + P_{[1]2}^2 = 30.4 + 1 = 31.4$$

$$C_{[2]2}^2 = \max(C_{[2]1}^2, C_{[1]2}^2) + P_{[2]2}^2 = 32.4 + 3.6 = 36$$

$$C_{[3]2}^2 = \max(C_{[3]1}^2, C_{[2]2}^2) + P_{[3]2}^2 = 36.6 + 5.1 = 41.7$$

In addition, the graph below presents the details of computing the completion times of three jobs on two machines and levels. Therefore, total completion time of job sequence  $(J_1, J_2, J_3)$  is  $\sum_{j=1}^3 C_{1[j]2}^2 = 109.1$



The graph of the completion times of three jobs on two machines and levels

For results on the problem without the learning effect, the reader may refer to 36-40.

### 3. The performances of four algorithms

The proposed problem is NP-hard since the classical total completion time minimization problem on  $m$ -machine flowshop without learning effect or reentrant consideration is NP-complete<sup>41</sup>. It becomes not economic to facilitate a searching procedure for the

optimal solution when the number of jobs is getting larger. Therefore, it has a good advantage policy to consider an effective heuristic or meta-heuristic for a good quality solution. Meanwhile, the literature indicates that a lot of researchers have paid more attention on constructing accurate algorithms within a limited amount of time for solving the classical total completion time minimization problem on m-machine flowshop problem such as Rajendran and Ziegler<sup>42</sup>, Woo and Yim<sup>43</sup>, and Framinan and Leisten<sup>44</sup>. Nawaz et al.<sup>45</sup> proposed a NEH heuristic for the makespan problem. Later, they modified the procedure for the flow time criterion. For simplification, it is referred to the four heuristics as RZ, WY, FL, and NEH, respectively. The main goals of this study is that in the first stage we first adopt four existed heuristics (the RZ, WY, FL, and NEH) for this problem; in the second stage we then use the same four methods combined with three local searches to solve the proposed problem; and in the last stage we develop a heuristic-based genetic algorithm seeded with four good different initials obtained from the second stage to find near-optimal solutions. Specifically, we are interested in investigating the impacts of learning effect and re-entrant times to the performances of all the proposed heuristics considered when the relative parameters are changed.

In what follows, we first evaluate the performance of the four heuristics proposed by Framinan and Leisten<sup>44</sup>, Nawaz et al.<sup>45</sup>, Woo and Yim<sup>43</sup>, and Rajendran and Ziegler<sup>42</sup> in solving the re-entrant permutation flowshop scheduling problem with a position-based learning effect to minimize the total completion time. The algorithms are coded in FORTRAN and run on a personal computer with an Intel(R) Core(TM)2 Quad CPU 2.66GHz with 4GB RAM.

In the tests the job processing times for each level are randomly generated from a uniform distribution [1, 100]. The number of jobs  $n$  is set to 8 and 10, the number of machines  $m$  at 2, 3, 5, and 10, and the number of re-entrant times  $l$  to 2, 3, and 5. The learning index  $a$  is also set to 70%, 80%, and 90%, which correspond to  $a = -0.515, -0.322, \text{ and } -0.152$ , respectively, according to Biskup's<sup>4</sup> model. In the tests, we first modify  $P_{ji} = \sum_{k=1}^l P_{ji}^k$  as the processing time of job  $J_j$  on machine  $M_i$ , and then apply the FL, NEH, WY, and RZ rule to yield the sequence. As a result, we tested a total of 72 cases with 100 instances in each case.

For the small number of jobs, we use full enumeration method to solve the instance to find an

optimal solution. We then compared the solutions obtained by all the heuristics or the GA algorithm in terms of the percentage error, given by

$$AEP = \frac{V_i - V^*}{V^*} \times 100\%$$

where  $V_i$  is the objective value obtained by algorithm  $H_i$  and  $V^*$  is the optimal objective value obtained by full enumeration. For each case, we recorded the mean percentage error. Tables 1-2 summarize the results.

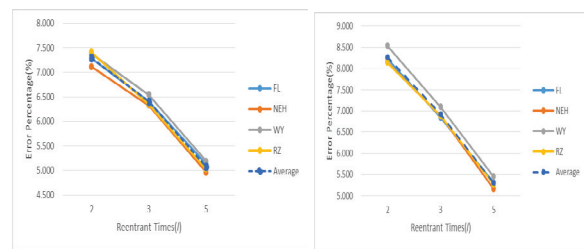


Fig. 1. The performance of four algorithms when  $l$  changes ( $n=8$  and 10)

As shown in Tables 1-2 and Figure 2, on an average, the mean percentage errors of the FL, NEH, WY, and RZ become large as the number of machines increases. On average, the mean error percentages of the FL, NEH, WY, and RZ are 6.273 and 6.791, 6.135 and 6.730, 6.370 and 7.026, and 6.262 and 6.758 at  $n=8$  and 10, respectively. The results show that NEH keeps slightly better than FL, WY, and RZ.

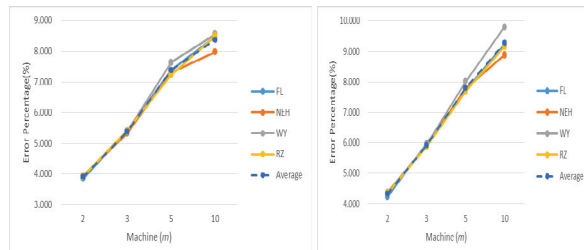


Fig. 2. The performance of four algorithms when  $m$  changes ( $n=8$  and 10)

As shown in Tables 1-2 and Figure 3, on an average, the mean percentage errors of the FL, NEH, WY, and RZ decline as the learning effect becomes weak. On average, the mean error percentages of the FL, NEH, WY, and RZ are 6.273 and 6.791, 6.135 and 6.730, 6.371 and 7.026, and 6.262 and 6.759 at  $n=8$  and 10, respectively. The results show that NEH still performs slightly better than FL, WY, and RZ.

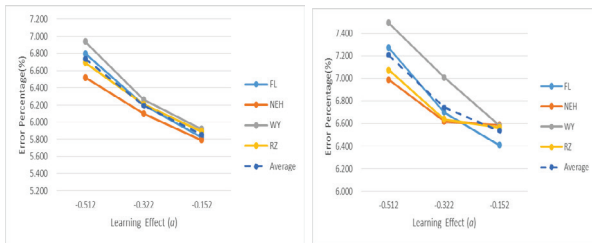


Fig. 3. The performance of four algorithms when  $a$  changes ( $n=8$  and  $10$ )

On average, NEH with mean error percentages of 6.135 and 6.730 performs slightly better than FL with 6.273 and 6.791, WY with 6.370 and 7.026, and RZ with 6.262 and 6.758 at  $n=8$  and  $10$  from Tables 1 and 2.

However, as mentioned in Demsar<sup>46</sup>, Garcia and Herrera<sup>47</sup>, the non-parametric tests should be preferred over the parametric ones in machine learning field. Therefore, some non-parametric statistical methods, such as the Friedman test, Iman and Davenport test, and post-hoc multiple comparison methods, such as, Nemenyi test, Holm procedure, and Shaffer static procedures<sup>47</sup>, are employed in analyzing the experimental data for this study. For more works involving with the experimental analysis in the field of evolutionary algorithms, we refer readers to<sup>48-53</sup>.

The numbers (1, 2, 3, 4) in parenthesis in the Tables 1 and 2 show the overall process of computation of average rankings of AEPs for the four algorithms. Friedman test, and Iman and Davenport tests<sup>47</sup> check whether the observed average ranks differ significantly from the mean rank  $R_j=2.5$ ,  $j=1, 2, 3, 4$ . Friedman test statistic, in this case, is 23.6787 (with chi-square distribution and degrees of freedom 3), while the critical value  $\chi^2_{0.05,3} = 7.815$ . Iman and Davenport test statistic, in this case, is 2.7566 (with  $F$  distribution and degrees of freedom (3,  $71 \cdot 3=213$ )), while the critical value  $F_{0.05,3,213} = 2.647$ .

Owing to both test statistics are greater than the respective critical values at the level of significance  $\alpha = 0.05$ , a post-hoc tests can be proceeded to detect significant pairwise differences among the search quality obtained from the four algorithms, FL, NEH, WY, and RZ. For this purpose, we have to compute the corresponding statistics and  $p$ -values. The test statistics, as Demsar<sup>46</sup>, for comparing the  $i$ -th and  $j$ -th algorithm is  $z = (R_i - R_j) / \sqrt{k(k+1)/(6N)}$  where  $R_j$  is the observed average rank computed through the Friedman test for the  $i$ -th algorithm,  $k=4$  is the number of algorithms to be compared, and  $N=72$  (that is  $N = n \cdot l \cdot m \cdot a = 2 \cdot 3 \cdot 4 \cdot 3$ ) is the number of parameter combinations (each parameter combination is treated as

a “block”) used in the comparison. The four observed  $R_j$ 's, computed in Table 2, for the FL, NEH, WY, and RZ are 2.243, 2.236, 3.132, and 2.389, respectively. Note that the standard error in the pairwise comparison between two algorithms is  $SE = \sqrt{k(k+1)/(6N)} = [(4 \cdot 5)/6 \cdot 72]^{1/2} = 0.2152$ .

Table 3 presents the family of hypotheses ordered by their  $p$ -values and the adjustments of  $\alpha$  by Nemenyi's ( $\alpha_{NM}$ ), Holm's ( $\alpha_{HM}$ ) and Shaffer's static ( $\alpha_{SH}$ ) procedures. In order to take multiple tests into account and to be compared directly with any chosen significance level  $\alpha$ , the corresponding adjusted  $p$ -value (APV) procedures, Nemenyi (APV<sub>NM</sub>), Holm (APV<sub>HM</sub>), and Shaffer's static (APV<sub>SH</sub>) are also employed in this study. The computing formulas refer to Garcia and Herrera (2008). These values are included in the last three columns of Table 3.

As shown in Table 3, it is obvious that Nemenyi's test, Holm's procedure and Shaffer's static procedure reject the hypotheses [ $i=1$  to  $i=3$ ] consistently. Since these three hypotheses all include the algorithm WY and the average rank of WY (3.312) is the largest among the four average ranks, it can be concluded that the algorithm WY performs worse than any of the other three algorithms does. Besides, the performance of three algorithms, NEK, FL and RZ, is not significant at the level of significance  $\alpha = 0.05$ . Moreover, the three APV's conform the same conclusion at  $\alpha = 0.05$ .

#### 4. The four algorithms combined with three local searches

It can be observed in Section 3 that on average the mean error percentage of all the four algorithms is more than 6% because of the re-entrant factor being brought up. In order to get a good quality solution, we build 12 heuristics by combining FL, NEH, WY, and RZ with three local search methods that are effective in treating the flowshop scheduling problem, namely the pairwise interchange, backward-shifted re-insertion, and forward-shifted re-insertion<sup>54</sup>. For example, the proposed heuristics in FL algorithm improved by the pairwise interchange, backward-shifted re-insertion, and forward-shifted re-insertion are recorded as FL\_PI, FL\_EBSR, and FL\_EFSR, respectively. The others are similar to the front. The tested results of all the proposed heuristics are reported in Tables 4-5.

It can be seen in Tables 4 and 5 that the heuristics with the pairwise interchange method outperform than those with backward-shifted re-insertion or forward-shifted re-insertion in terms of the numbers of cases,

i.e., 124 cases for PI while 3 and 17 cases for backward-shifted re-insertion and forward-shifted re-insertion at  $n=8$ , while 136 cases for PI while 2 and 6 cases for backward-shifted re-insertion and forward-shifted re-insertion at  $n=10$ . On an average, among four heuristics, FL algorithm improved by PI method is slightly better than those with backward-shifted re-insertion and forward-shifted re-insertion in terms of mean error percentage. So do the NEH, WY, and RZ algorithms in Tables 4 and 5. On average, the results indicated that FL improved by PI method with the mean error percentages of 2.436 at  $n=8$  and 2.993 at  $n=10$  are smaller than each of other three algorithms improved by PI method.

### 5. Four heuristic-based genetic algorithms

As stated in the previous section, FL improved by PI method with the average mean error of less than 3% outperforms the other three algorithms with PI method. However, it can still be seen in Table 4 that some case has a bit bigger mean error percentage of 4.649. In view of this situation, we attempt to use a heuristic-based genetic algorithm further to improve near-optimal solutions. Literature releases that the genetic algorithm (GA) has been widely and successfully used in many combinational problems<sup>55-61</sup>. It can be seen that the initial populations are generated randomly in GA settings<sup>62</sup>. However, some research suggests that it is a good idea to construct the initial population based on other heuristics that makes GA to yield the good quality solution more quickly<sup>63-66</sup>. In light of this observation, this study will adopt four heuristics as initial sequences in GA algorithm which are included FL\_PI, NEH\_PI, WY\_PI, and RZ\_PI. The main steps of the proposed GA are summarized in the following.

#### Genetic algorithm process:

- Step 1:** Representation of structure- a sequence of the jobs presents a structure in the problem that is followed by Etiler *et al.*<sup>65</sup>.
- Step 2:** Initial population- We apply four heuristics as initial sequences in GA. In first one, FL algorithm is improved by PI method which is recorded as FL\_PI+GA. The second, third, and fourth genetic algorithms, denoted as NEH\_PI+GA, WY\_PI+GA, and RZ\_PI+GA, are the NEH, WY, and RZ algorithms improved by PI method, respectively.
- Step 3:** Population size- We adopt an initial population and create other members by using an interchange mutation operator up to population

size<sup>64</sup>. In a preliminary trial, the population size  $N$  is set at 20 in our computational experiment.

- Step 4:** Fitness function- The fitness function of the strings can be calculated as

$$g(S_i(v)) = \max_{1 \leq k \leq N} \left\{ \sum_{j=1}^n C_{jm}^R(S_k(v)) \right\} - \sum_{j=1}^n C_{jm}^R(S_i(v))$$

where  $S_i(v)$  is the  $i$ th string chromosome in the  $v$ -th generation,  $\sum_{j=1}^n C_{jm}^R(S_i(v))$  is the total completion time of  $S_i(v)$ , and  $g(S_i(v))$  is the fitness function of  $S_i(v)$ . The probability,  $P(S_i(v))$ , of selection for a schedule can be calculated as

$$P(S_i(v)) = g(S_i(v)) / \sum_{k=1}^N g(S_k(v))$$

This is also the criterion used for the selection of parents for the reproduction of children.

- Step 5:** Crossover- Following by Falkenauer and Bouffouix<sup>67</sup>, Etiler and Toklu<sup>68</sup>, Etiler *et al.*<sup>65</sup>, we adopt linear order crossover (LOX) method in each GA. In order to protect the best schedule which has the minimum total completion time at each generation, we transfer this schedule to the next population with no change, or the crossover rate  $P_c = 1$ .
- Step 6:** Mutation- The mutation rates ( $P_m$ ) are set at 0.25 based on our preliminary experiment.
- Step 7:** Selection- In each GA, the population sizes are set to 20 from generation to generation. Excluding the best schedule which has the minimum total completion time, the other offsprings are generated randomly from the parent chromosomes by the roulette wheel method.
- Step 8:** Stop rule- Each GA is terminated after 1000 generations in our preliminary experiment.

We conducted computational experiments to test the performance of the four meta\_heuristics FL\_PI+GA, NEH\_PI+GA, WY\_PI+GA, and RZ\_PI+GA in solving the re-entrant permutation flowshop scheduling problem by using the same data. All results are reported in Tables 6 and 7.

As shown in Tables 6-7 and Fig. 4, on an average, the mean percentage errors of FL\_PI+GA, NEH\_PI+GA, WY\_PI+GA, and RZ\_PI+GA decline as the reentrant number becomes larger. On average, the mean error percentages of FL\_PI+GA, NEH\_PI+GA, WY\_PI+GA, and RZ\_PI+GA are only 1.048 and 1.904, 1.109 and 1.845, 1.112 and 1.893, and 1.040 and 1.893



at  $n=8$  and  $10$ , respectively. The results show that FL\_PI+GA and RZ\_PI+GA are only slightly better than NEH\_PI+GA and WY\_PI+GA. So do the similar trend for the performances when changing the numbers of machines or the learning effect. In addition, it can be observed in Tables 6-7 and Fig. 4 that the mean error percentage can be declined sharply by these improvements for four algorithms no matter what the re-entrant time or number of machine or learning effect changes.

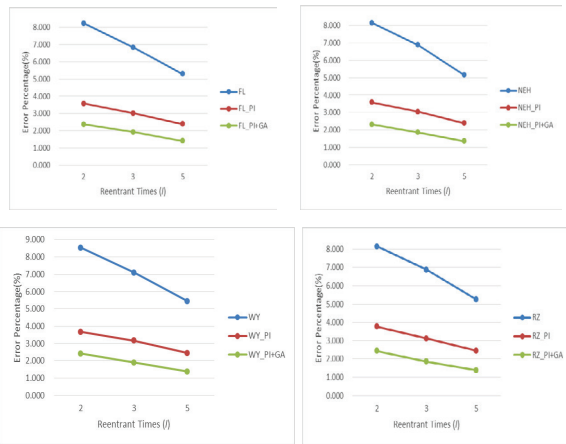


Fig. 4. The performance of four and their improved algorithms when  $l$  changes ( $n=10$ )

### 6. Four heuristic-based genetic algorithms on the larger number of job

In the tests we randomly generated the job processing time for each level from a uniform distribution [1, 100]. We set the number of machines  $m$  at 2, 3, 5, and 10, and the number of re-entrant times  $l$  at 2, 3, and 5. We also set the learning index  $a$  at 70%, 80%, and 90%. As a result, we tested a total of 72 cases with 100 instances in each case.

For the cases with a large number of jobs, i.e.,  $n = 50$  and  $100$ , we first solved each instance by using the eight heuristics FL\_PI, FL\_PI+GA, NEH\_PI, NEH\_PI+GA, WY\_PI, WY\_PI+GA, RZ\_PI, and RZ\_PI+GA algorithms to obtain near-optimal solutions. We measure the performance of a heuristic or an GA algorithm in terms of the relative percentage deviation (RPD), defined as

$$RPD = \frac{V - V_{\min}}{V_{\min}} \times 100\%$$

where  $V$  is the objective value obtained by a heuristic or a GA and  $V_{\min}$  is the best solution among FL\_PI,

FL\_PI+GA, NEH\_PI, NEH\_PI+GA, WY\_PI, WY\_PI+GA, RZ\_PI, and RZ\_PI+GA algorithms. For each case, we recorded the average (denoted as RPD) percentage deviation. Tables 8-9 summarize the results. We see from Tables 8 and 9 that the RPDs of FL\_PI+GA, NEH\_PI+GA, WY\_PI+GA, and RZ\_PI+GA have a decreasing trend when the re-entrant time increases as shown in Fig. 5, but have an increasing trend when the number of machines becomes big or the learning effect gets stronger as shown in Fig. 6 and 7. The results indicate that FL\_PI+GA performs better than the other three heuristics regardless of the number of re-entrant times or machines or learning effect. In addition, there is no clearly dominance relationship between NEH\_PI+GA and WY\_PI+GA from Figs 5-7.

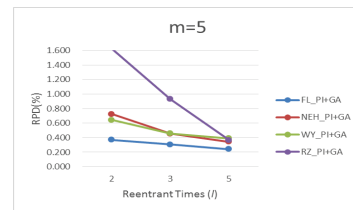


Fig 5. The performance of four GA algorithms when  $l$  changes ( $n=100, m=5$ )

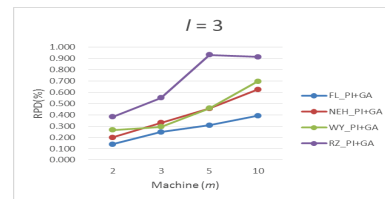


Fig. 6. The performance of four GA algorithms when  $m$  changes ( $n=100, l=3$ )

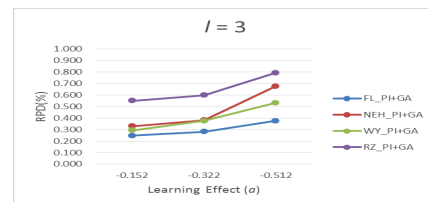


Fig. 7. The performance of four GA algorithms when  $a$  changes ( $n=100, l=3$ )

Furthermore, from Tables 8-9, it can be observed that on average FL\_PI+GA has smaller RPD values of 0.465% at  $n = 50$  and 0.327% at  $n = 100$ , which are better than those of NEH\_PI+GA of 0.646% and 0.530%, of WY\_PI+GA of 0.666% and 0.633%, and of RZ\_PI+GA of 0.829% and 0.769% at  $n = 50$  and  $100$ , respectively. And, Fig. 8 releases that the RPD of FL\_PI+GA has the least dispersion among the four GA algorithms.

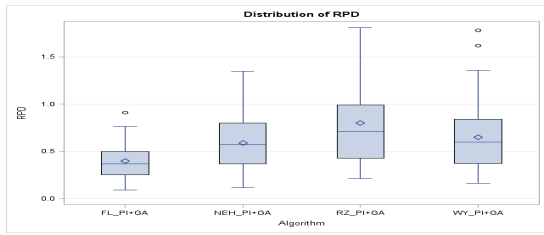


Fig 8. The RPD distributions of four GAs (n=50, 100)

As shown in Fig. 9, Box plots convey the information contained in Tables 6 and 7. It is obviously shown the difference between search results for the twelve algorithms constructed in the three stages. The applied heuristics, especially the GAs used in the third stage, improve considerably the quality of the search solutions. Overall, it is concluded based on the above discussions that the FL\_PI+GA algorithm does well and it might be adopted to solve the proposed problem in view of its stability and its ability in producing near-optimal solutions.

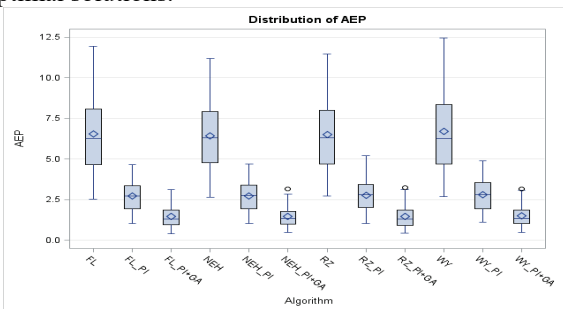


Fig 9. AEP distributions of twelve algorithms (n=8, 10)

However, as done in Section 3, we employ non-parametric statistical tests to check whether there exists any significant difference among the performances of the four GA-based algorithms for the large size of jobs. The numbers (1, 2, 3, 4) in parenthesis in Tables 8 and 9 show the overall process of computation of average rankings of RPDs for the four GA-based algorithms. The four observed  $R_j$ 's, computed in Table 9, for the FL\_PI+GA (FGA), NEH\_PI+GA (NGA), WY\_PI+GA (WGA), and RZ\_PI+GA (RGA) are 1.153, 2.500, 3.403, and 2.944, respectively. The notations and formulas are omitted since they are the same as those used in the Section 3.

Friedman test, and Iman and Davenport tests check whether the observed average ranks differ significantly from the mean rank  $R_j=2.5$ ,  $j=1, 2, 3, 4$ . Friedman test statistic, in this case, is 122.15 and the correspondent critical value, at  $\alpha=0.05$ , is  $\chi^2_{0.05;3} = 7.815$ . Iman and Davenport test statistic, in this case, is 95.46 and the

correspondent critical value is  $F_{0.05,3,213} = 2.647$ . Given these test statistics are greater than the respective critical values, there are significant differences among the observed results from the four GA-based algorithms. Thus, a post-hoc statistical analysis could help to detect specific differences among the search quality obtained from the four GA-based algorithms.

As shown in Table 10, it can be seen that Nemenyi's test rejects the hypotheses [ $i=1$  to  $i=4$ ], Holm's procedure and the Shaffer's static reject the hypotheses [ $i=1$  to  $i=4$ , and  $i=6$ ]. Besides, All three APVs reject the hypotheses [ $i=1$  to  $i=4$ ] at level of significance 0.05. Therefore, it can be concluded that the algorithm FL\_PI+GA (FGA) performs significantly better than any of other three GA-based algorithms does, and NEH\_PI+GA (NGA) performs significantly better than WY\_PI+GA (WGA) does.

### 7. Conclusions

In this study we address the re-entrant permutation flowshop scheduling problem with learning considerations to minimize the total completion time. Given the intractability of the problem, we build four heuristics based on three local search methods to approximately solve the problem. Moreover, in order to find good quality near-optimal solutions, we apply genetic algorithm using the solutions obtained by the heuristics as initial solutions to solve the problem. The computational results show that the GA algorithm seeded with the solution obtained by combining FL algorithm improved by interchange method performs best in terms of solution quality. Future research may consider including the due dates to solve the problem involving other scheduling objectives such as the total tardiness.

### Acknowledgements

The authors are grateful to the Editor and anonymous referees for their many helpful comments on the earlier versions of our paper. This paper was supported by the NSC of Taiwan under grant numbers MOST 105-2221-E-035-053-MY3 and MOST 103-2410-H-035-022-MY2.



## References

1. M. Pinedo, Scheduling: theory, algorithms and systems. NJ, Prentice-Hall: Upper Saddle River, (2008).
2. Y. Yin, T.C.E. Cheng, D.-J. Wang, C.-C. Wu, (2015) Improved algorithms for single-machine serial-batch scheduling with rejection to minimize total completion time and total rejection cost, *IEEE Transactions on Systems, Man and Cybernetics: Systems*, (2015) doi: 10.1109/TSMC.2015.2505644.
3. Y. Yin, J. Xu, T.C.E. Cheng, C.-C. Wu, D.-J. Wang, Approximation schemes for single-machine scheduling with a fixed maintenance activity to minimize the total amount of late work, *Naval Research Logistics*, 63 (2016) 172-183.
4. D. Biskup, Single-machine scheduling with learning considerations, *European Journal of Operational Research*, 115 (1999) 173-178.
5. D. Nadler, W.D. Smith, Manufacturing progress functions for types of processes, *International Journal of Production Research*, 2 (1963) 115-135.
6. L.E. Yelle, The learning curve: historical review and comprehensive survey, *Decision Science*, 10 (1979) 302-328.
7. P. Higgins, P. Le Roy, L. Tierney, Manufacturing Planning and Control-Beyond MRP II. Chapman & Hall, London (1996).
8. T.C.E. Cheng G. Wang, Single machine scheduling with learning effect considerations, *Annals of Operations Research*, 98 (2000) 273-290.
9. D. Biskup, A state-of-the-art review on scheduling with learning effect, *European Journal of Operational Research*, 188 (2008) 315-329.
10. S.J. Yang, C.J. Hsu, D.L. Yang, Parallel-machine scheduling with setup and removal times under consideration of the learning effect, *Journal of the Chinese Institute of Industrial Engineers*, 27(5) (2010) 372-378.
11. Y. Yin, D. Xu, J. Wang, Single-machine scheduling with a general sum-of-actual-processing-times-based and job-position-based learning effect, *Applied Mathematical Modelling*, 34(11) (2010) 3623-3630.
12. N. Yin, X.-Y. Wang, Single machine scheduling with controllable processing times and learning effect, *International Journal of Advanced Manufacturing Technology*, 54 (2011) 743-748.
13. J.-B. Wang, Single-machine scheduling with a sum-of-actual- processing- time- based learning effect, *Journal of the Operational Research Society*, 61 (2010) 172-177.
14. S.-J. Yang, D.-L. Yang, Scheduling problems with past-sequence- dependent delivery times and learning effects, *Journal of the Operational Research Society*, 63 (2012) 1508-1515.
15. J.-B. Wang, Q. Guo, A due-date assignment problem with learning effect and deteriorating jobs, *Applied Mathematical Modelling*, 34, (2010) 309-313.
16. X.-Y. Wang, Z. Zhou, X. Zhang, P. Ji, J.-B. Wang, several flow shop scheduling problems with truncated position-based learning effect, *Computers & Operations Research*, 40 (2013) 2906-2929.
17. J.-B. Wang, Z.Q. Xia, Flow-shop scheduling with a learning effect, *Journal of the Operational Research Society*, 56 (2005) 1325-1330.
18. C.-C. Wu, W.-C. Lee, A note on the total completion time problem in a permutation flowshop with a learning effect, *European Journal of Operational Research*, 192 (2009) 343-347.
19. J.-B. Wang, M.Z. Wang, Worst-case analysis for flow shop scheduling problems with an exponential learning effect, *Journal of the Operational Research Society*, 63 (2012) 130-137.
20. Y.-H. Chung, L.-I. Tong, Bi-criteria minimization for the permutation flowshop scheduling problem with machine-based learning effects, *Computers & Industrial Engineering*, 63(1) (2012) 302-312.
21. W.-H. Kuo, C.-J. Hsu, D.-L. Yang, Worst-case and numerical analysis of heuristic algorithms for flowshop scheduling problems with a time-dependent learning effect, *Information Sciences*, 184(1) (2012) 282-297.
22. T.C.E. Cheng, C.-C. Wu, J.-C. Chen, W.-H. Wu, S.-R. Cheng, Two-machine flowshop scheduling with a truncated learning function to minimize the makespan, *International Journal of Production Economics*, 141(1) (2013) 79-86.
23. M. Cheng, Flowshop scheduling problems with a position-dependent exponential learning effect, *Mathematical Problems in Engineering* (2013) doi. org/ 10.1155/ 2013/ 753123.
24. L.H. Sun, K. Cui, J.-H. Chen, J. Wang, X.C. He, Some results of the worst-case analysis for flow shop scheduling with a learning effect, *Annals of Operations Research*, 211(2013) 481-490.
25. L.-H. Sun, K. Cui, J.-H. Chen, J. Wang, X.-C. He, Research on permutation flow shop scheduling problems with general position-dependent learning effects, *Annals of Operations Research*, 211 (2013) 473-480.
26. A. Janiak, T. Krysiak, R. Trela, Scheduling problems with learning and ageing effects: a survey, *Decision Making in Manufacturing*, 5(1-2) (2011) 19-36.
27. F.D. Vargas-Villamil, D.E. Rivera, A model predictive control approach for real-time optimization of reentrant manufacturing lines, *Computers in Industry*, 45 (2001) 45-57.
28. M.Y. Wang, S.P. Sethi, S.L. van de Velde, Minimizing makespan in a class of reentrant shops, *Operations Research*, 45 (1997) 702-712.
29. G. Bengu, A simulation-based scheduler for flexible flowlines, *International Journal of Production Research*, 32 (1994) 321-344.
30. C.F. Bispo, S. Tayur, Managing simple re-entrant flow lines: Theoretical foundation and experimental results, *IIE Transaction*, 33 (2001) 609-623.

31. W. Kubiak, S.X.C. Lou, Y. Wang, Mean flow time minimization in reentrant job-shops with a hub, *Operations Research*, 44 (1996) 764-776.
32. R. Uzsoy, C.Y. Lee, L.A. Martin-Vega, A review of production planning and scheduling models in the semiconductor industry—Part I: System characteristics, performance evaluation and production planning, *IIE Transaction*, 24 (1992) 47-60.
33. R. Bellman, R. Ernest, *Mathematical Aspects of Scheduling and Applications*. Oxford: Pergamon Press, (1982).
34. D. Lin, C.K.M. Lee, A review of the research methodology for the re-entrant scheduling problem, *International Journal of Production Research*, 49(8) (2011) 2221-2242.
35. C.-C. Wu, Y.Q. Yin, T.C.E. Cheng, S.-Y. Liu, W.-H. Wu, Re-entrant flowshop scheduling with learning considerations to minimize the makespan, *Iranian Journal of Science and Technology, Transactions A: Science, under review*, (2016).
36. J.C.H. Pan, J.-S. Chen, Minimizing makespan in re-entrant permutation flow-shops, *Journal of the Operational Research Society*, 54 (2003) 642-653.
37. J.M. Wilson, Alternative formulations of a flow-shop scheduling problem, *Journal of the Operational Research Society*, 40 (1989) 395-399.
38. J.-S. Chen, A branch-and-bound procedure for the reentrant permutation flow-shop scheduling problem, *International Journal of Advanced Manufacturing Technology*, 29 (2006) 1186-1193.
39. J.-S. Chen, J.C.H. Pan, C.K. Wu, Minimizing makespan in reentrant flow-shops using hybrid tabu search, *International Journal of Advanced Manufacturing Technology*, 34 (2007) 353-361.
40. J. Xu, Y. Yin, T.C.E. Cheng, C.-C. Wu, S. Gu, A memetic algorithm for the re-entrant permutation flowshop scheduling problem to minimize the makespan, *Applied Soft Computing*, 24 (2014) 277-283.
41. J.K. Lenstra, A.H.G. Rinnooy Kan, P. Brucker, Complexity of machine scheduling problems, *Annals of Operations Research*, 1 (1977) 343-362.
42. C. Rajendran, H. Ziegler, An efficient heuristic for scheduling in a flowshop to minimize total weighted flowtime of jobs, *European Journal of Operational Research*, 103 (1997) 129-138.
43. D.S. Woo, H.S. Yim, A heuristic algorithm for mean flowtime objective in scheduling, *Computers and Operations Research*, 25 (1998) 175-182.
44. J.M. Framinan, R. Leisten, An efficient constructive heuristic for flowtime minimisation in permutation flow shops, *Omega: The International Journal of Management Science*, 31 (2003) 311-317.
45. M. Nawaz, E.E. Enscore, I. Ham, A heuristic algorithm for the  $m$ -machine,  $n$ -job sequencing problem, *Omega: The International Journal of Management Science*, 11 (1983) 91-95.
46. J. Demšar, Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research*, 7 (2006) 1-30.
47. S. García, F. Herrera, An extension on Statistical Comparisons of Classifiers over Multiple Data Sets for all pairwise comparisons, *Journal of Machine Learning Research*, 9 (2008) 2677-2694.
48. S. García, A. Fernández, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability, *Soft Computing*, 13 (2009) 959-977.
49. S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter, *Optimization, Journal of Heuristics*, 15 (2009) 617-644.
50. S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, *Information Sciences*, 180 (2010) 2044-2064.
51. J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm and Evolutionary Computation*, 1 (2011) 3-18.
52. J. Derrac, S. García, S. Hui, P.N. Suganthan, F. Herrera, Analyzing convergence performance of evolutionary algorithms: A statistical approach, *Information Sciences*, 289 (2014) 41-58.
53. J. Luengo, S. García, F. Herrera, A study on the use of statistical tests for experimentation with neural networks: Analysis of parametric test conditions and non-parametric tests, *Expert Systems with Applications*, 36 (2009) 7798-7808.
54. F. Della Croce, V. Narayan, R. Tadei, The two-machine total completion time flow shop problem, *European Journal of Operational Research*, 90 (1996) 227-237.
55. M.A. Ahmadi, S.R., Shadizadeh, RETRACTED ARTICLE: Intelligent approach for prediction of minimum miscible pressure by evolving genetic algorithm and neural network, *Neural Computing and Applications*, 23(2) (2013) 569-569.
56. S.H. Cheng, M.C. Chen, Y.C. Liou, Artificial chromosomes with genetic algorithm 2 (ACGA2) for single machine scheduling problems with sequence-dependent setup times, *Applied Soft Computing*, 17 (2014) 167-175.
57. O.J. Mengshoel, S.F. Galán, A. de Dios, Adaptive generalized crowding for genetic algorithms, *Information Sciences*, 258 (2014) 140-159.
58. F. Tao, Y. Feng, L. Zhang, T.W. Liao, CLPS-GA: A case library and Pareto solution-based hybrid genetic algorithm for energy-aware cloud service scheduling, *Applied Soft Computing*, 19 (2014) 264-279.
59. F. Valdez, P. Melin, O. Castillo, Modular Neural Networks architecture optimization with a new nature

- inspired method using a fuzzy combination of Particle Swarm Optimization and genetic algorithms, *Information Sciences*, 270 (2014) 143-153.
60. Y. Xu, K., Li, J., Hu, K. Li, A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues, *Information Sciences*, 270 (2014) 255-287.
  61. C.-C. Wu, Y. Yin, W.H. Wu, H.M. Chen, S.R. Cheng, Using a branch-and-bound and a genetic algorithm for a single-machine total late work scheduling problem, *Soft Computing*, 20 (2016) 1329-1339.
  62. J.C. Bean, Genetic algorithms and random keys for sequencing and optimization, *ORSA Journal of Computing*, 6 (1964) 154-160.
  63. C. Reeves, Heuristics for scheduling a single machine subject to unequal job release times, *European Journal of Operational Research*, 80 (1995) 397-403.
  64. C.L. Chen, V.S. Vempati, N. Aljaber, An application of genetic algorithms for flow shop problems, *European Journal of Operations Research*, 80 (1995) 389-396.
  65. O. Etiler, B. Toklu, M. Atak, J. Wilson, (2004) A generic algorithm for flow shop scheduling problems, *Journal of Operations Research Society*, 55(8) (2004) 830-835.
  66. C.-C. Wu, P.-H. Hsu, J.-C. Chen, N.-S. Wang, Genetic algorithm for minimizing the total weighted completion time scheduling problem with learning and release times, *Computers & Operations Research*, 38(7) (2011)1025-1034.
  67. E. Falkenauer, S. Bouffoix, A genetic algorithm for job shop. Proceedings of the 1991 IEEE International Conference on Robotics and Automation, (1991).
  68. O. Etiler, B. Toklu, Comparison of genetic crossover operators using in scheduling problems, *Journal Inst Technology, Gazi University, Turkey*, 14 (2001) 21-32.

Appendix: Tables

Table 1. The performance of four heuristics (n=8)

<i>l</i>	<i>m</i>	<i>a</i>	Heuristic algorithms				
			FL	NEH	WY	RZ	
			Mean Error Percentage (%)				
2	2	-0.152	5.535 (4)	5.448 (2)	5.302 (1)	5.516 (3)	
		-0.322	4.865 (1)	4.958 (2)	4.962 (3)	5.176 (4)	
		-0.512	4.577 (1)	5.040 (4)	4.678 (2)	4.727 (3)	
	3	-0.152	6.805 (2)	6.844 (3)	6.720 (1)	7.271 (4)	
		-0.322	6.591 (2)	6.602 (3)	6.293 (1)	6.821 (4)	
		-0.512	6.385 (2)	6.617 (3)	6.795 (4)	6.279 (1)	
		5	-0.152	8.731 (3)	8.403 (1)	9.226 (4)	8.639 (2)
			-0.322	8.675 (1)	8.812 (2)	9.268 (4)	8.833 (3)
			-0.512	9.184 (4)	8.352 (2)	8.956 (3)	8.302 (1)
	10	-0.152	7.309 (2)	7.283 (1)	7.578 (4)	7.550 (3)	
		-0.322	8.753 (3)	7.843 (1)	8.534 (2)	9.207 (4)	
		-0.512	10.015 (2)	9.209 (1)	10.235 (3)	10.637 (4)	
		3	-0.152	4.100 (4)	4.067 (2)	4.074 (3)	3.978 (1)
			-0.322	3.761 (2)	3.790 (3)	3.978 (4)	3.677 (1)
			-0.512	3.668 (1)	3.822 (3)	3.870 (4)	3.761 (2)
3	-0.152	5.840 (4)	5.417 (1)	5.776 (3)	5.702 (2)		
	-0.322	5.484 (4)	5.174 (1)	5.443 (3)	5.278 (2)		
	-0.512	5.776 (3.5)	5.615 (2)	5.776 (3.5)	5.416 (1)		
	5	-0.152	6.622 (1)	6.829 (3)	6.977 (4)	6.827 (2)	
		-0.322	7.247 (2)	7.518 (4)	7.278 (3)	6.926 (1)	
		-0.512	7.878 (2)	7.841 (1)	8.412 (4)	7.939 (3)	
10	-0.152	6.487 (1)	7.077 (4)	6.959 (3)	6.723 (2)		
	-0.322	8.839 (3)	8.426 (1)	8.995 (4)	8.598 (2)		
	-0.512	11.162 (3)	10.371 (1)	10.982 (2)	11.344 (4)		
	5	-0.152	2.713 (2)	2.755 (3)	2.702 (1)	2.872 (4)	
		-0.322	2.544 (1)	2.629 (2)	2.676 (3)	2.724 (4)	
		-0.512	2.810 (2)	2.927 (3)	3.012 (4)	2.780 (1)	
3	-0.152	3.962 (4)	3.782 (1)	3.871 (3)	3.840 (2)		
	-0.322	3.858 (2)	3.933 (4)	3.765 (1)	3.921 (3)		
	-0.512	3.874 (1)	3.917 (2)	3.939 (3)	4.142 (4)		
	5	-0.152 ( )	5.611 (3)	5.533 (2)	5.869 (4)	5.381 (1)	
		-0.322	5.739 (2)	6.029 (4)	5.881 (3)	5.586 (1)	
		-0.512	6.766 (3)	6.448 (1)	6.833 (4)	6.610 (2)	
10	-0.152	6.112 (3)	6.035 (2)	5.955 (1)	6.382 (4)		
	-0.322	8.089 (4)	7.454 (1)	8.036 (3)	7.704 (2)		
	-0.512	9.469 (3)	8.075 (1)	9.746 (4)	8.370 (2)		
Average			6.273	6.135	6.370	6.262	

**Table 2. The performance of four heuristics (n=10)**

<i>l</i>	<i>m</i>	<i>a</i>	Heuristic algorithms			
			FL	NEH	WY	RZ
			Mean Error Percentage (%)			
2	2	-0.152	5.739 (1)	5.937 (3)	5.936 (2)	6.023 (4)
		-0.322	5.645 (2)	5.752 (4)	5.672 (3)	5.562 (1)
		-0.512	5.132 (3)	5.056 (1)	5.121 (2)	5.253 (4)
	3	-0.152	7.748 (3)	7.720 (2)	7.382 (1)	7.839 (4)
		-0.322	7.432 (3)	7.322 (2)	7.532 (4)	7.252 (1)
		-0.512	7.734 (4)	7.342 (3)	7.227 (2)	6.963 (1)
	5	-0.152	9.420 (1)	9.585 (2)	9.776 (4)	9.730 (3)
		-0.322	9.691 (3)	9.559 (1)	10.159 (4)	9.635 (2)
		-0.512	9.703 (2)	9.858 (3)	10.511 (4)	8.908 (1)
	10	-0.152	8.587 (1)	9.182 (2)	9.314 (4)	9.196 (3)
		-0.322	10.040 (3)	9.735 (1)	11.367 (4)	9.969 (2)
		-0.512	11.959 (3)	10.754 (1)	12.452 (4)	11.313 (2)
3	2	-0.152	4.330 (1)	4.610 (3)	4.644 (4)	4.609 (2)
		-0.322	4.188 (1)	4.345 (3)	4.336 (2)	4.408 (4)
		-0.512	4.029 (1)	4.333 (3)	4.343 (4)	4.137 (2)
	3	-0.152	6.119 (2)	6.211 (3)	6.285 (4)	6.074 (1)
		-0.322	5.547 (1)	6.035 (3)	6.042 (4)	5.829 (2)
		-0.512	5.779 (3)	5.456 (1)	5.717 (2)	5.824 (4)
	5	-0.152	7.886 (2)	8.269 (4)	8.021 (3)	7.881 (1)
		-0.322	7.457 (2)	7.559 (3)	7.566 (4)	7.318 (1)
		-0.512	8.060 (2)	7.885 (1)	8.274 (4)	8.096 (3)
	10	-0.152	7.712 (1)	7.932 (4)	7.907 (3)	7.893 (2)
		-0.322	9.509 (3)	8.783 (1)	9.822 (4)	8.989 (2)
		-0.512	11.467 (2)	11.201 (1)	12.167 (4)	11.488 (3)
5	2	-0.152	2.882 (1)	2.925 (3)	2.905 (2)	2.980 (4)
		-0.322	2.841 (1)	3.044 (4)	3.026 (3)	2.867 (2)
		-0.512	3.195 (1)	3.336 (4)	3.272 (3)	3.212 (2)
	3	-0.152	4.223 (2)	4.280 (4)	4.250 (3)	4.152 (1)
		-0.322	4.448 (4)	4.299 (2)	4.325 (3)	4.219 (1)
		-0.512	4.705 (4)	4.389 (1)	4.704 (3)	4.653 (2)
	5	-0.152	5.617 (1)	5.694 (3)	5.648 (2)	5.705 (4)
		-0.322	5.619 (1)	5.634 (2)	5.938 (4)	5.647 (3)
		-0.512	6.121 (2)	6.013 (1)	6.191 (4)	6.170 (3)
	10	-0.152	6.611 (1)	6.709 (3)	6.904 (4)	6.704 (2)
		-0.322	7.961 (3)	7.342 (1)	8.304 (4)	7.949 (2)
		-0.512	9.352 (3)	8.223 (1)	9.911 (4)	8.865 (2)
Average			6.791 (2.243)	6.730 (2.236)	7.026 (3.132)	6.586 (2.389)

**Table 3. Family of hypotheses ordered by *p*-values and other information to conduct pairwise comparisons for small size of jobs, initial  $\alpha=0.05$ .**

<i>i</i>	hypothesis	<i>z</i>	<i>p</i> -value	$\alpha_{NM}$	$\alpha_{HM}$	$\alpha_{SH}$	APV <sub>NM</sub>	APV <sub>HM</sub>	APV <sub>SH</sub>
1	WY vs. NEH	4.1635	0.00003	0.0083	0.0083	0.0083	0.00018	0.00018	0.00018
2	WY vs. FL	4.1312	0.00004	0.0083	0.01	0.0167	0.00024	0.00020	0.00012
3	WY vs. RZ	3.4534	0.0006	0.0083	0.0125	0.0167	0.00360	0.00240	0.00180
4	RZ vs. NEH	0.7100	0.4777	0.0083	0.0167	0.0167	1	1	1
5	RZ vs. FL	0.6778	0.4979	0.0083	0.025	0.025	1	1	1
6	FL vs. NEH	0.0323	0.9742	0.0083	0.05	0.05	1	1	1

**Table 4. The performance of improved heuristics (n=8)**

			Heuristic algorithms											
			FL_EFSR	FL_EBSR	FL_PI	NEH_EFSR	NEH_EBSR	NEH_PI	WY_EFSR	WY_EBSR	WY_PI	RZ_EFSR	RZ_EBSR	RZ_PI
<i>l</i>	<i>m</i>	<i>a</i>	Mean Error Percentage(%)											
2	2	-0.152	2.464	2.475	2.054	2.438	2.419	2.047	2.349	2.400	1.935	2.527	2.574	2.081
		-0.322	2.159	1.898	1.536	2.314	1.977	1.684	2.254	1.964	1.715	2.422	2.269	1.956
		-0.512	2.042	1.452	1.441	2.376	1.634	1.643	2.207	1.534	1.625	2.047	1.629	1.517
3	3	-0.152	2.992	3.228	2.771	3.324	3.413	3.106	3.380	3.187	2.944	3.359	3.378	2.982
		-0.322	2.904	2.833	2.490	3.149	3.016	2.805	2.966	2.722	2.625	3.248	2.960	2.712
		-0.512	2.892	2.393	3.267	3.220	2.559	2.467	3.408	2.692	2.598	3.031	2.602	2.446
5	5	-0.152	4.101	4.155	3.656	4.065	3.856	3.749	4.245	4.517	3.829	4.266	4.177	3.696
		-0.322	3.966	3.957	3.596	3.910	3.745	3.464	4.162	4.177	3.987	4.126	4.119	3.724
		-0.512	4.145	3.689	3.309	3.471	3.146	3.094	3.789	3.613	3.572	3.887	3.704	3.348
10	10	-0.152	3.971	3.521	3.127	4.085	3.569	3.462	3.934	3.806	3.177	4.378	3.700	3.299
		-0.322	4.032	4.212	3.323	3.941	3.545	3.332	4.156	4.329	3.203	4.194	4.100	3.166
		-0.512	3.755	4.311	3.150	3.890	3.758	3.299	4.038	4.662	3.219	3.689	4.530	3.218
3	2	-0.152	2.085	2.025	1.620	2.115	1.948	1.628	2.066	1.978	1.666	1.965	1.888	1.541
		-0.322	1.812	1.581	1.436	1.889	1.553	1.569	1.927	1.635	1.599	1.753	1.620	1.331
		-0.512	1.649	1.260	1.321	1.797	1.324	1.429	1.851	1.386	1.425	1.696	1.333	1.299
3	3	-0.152	2.854	3.017	2.648	2.592	2.819	2.452	2.775	2.971	2.601	2.705	2.840	2.496
		-0.322	2.735	2.663	2.360	2.727	2.449	2.269	2.767	2.566	2.277	2.562	2.272	2.021
		-0.512	2.822	2.215	2.353	2.876	2.253	2.148	3.022	2.349	2.188	2.736	2.068	2.033
5	5	-0.152	3.505	3.419	3.249	3.675	3.530	3.297	3.604	3.564	3.292	3.455	3.455	3.110
		-0.322	3.386	3.433	3.242	3.693	3.511	3.237	3.539	3.403	3.094	3.258	3.400	3.049
		-0.512	3.256	3.149	2.830	3.770	3.301	3.176	3.857	3.408	3.181	2.997	3.242	3.001
10	10	-0.152	3.196	3.454	2.998	3.686	3.624	3.199	3.591	3.494	3.046	3.428	3.696	3.125
		-0.322	3.528	4.050	3.306	3.756	3.444	3.291	3.824	3.842	3.517	3.577	3.814	3.233
		-0.512	3.664	3.530	3.445	3.908	3.023	3.034	4.004	3.571	3.349	3.461	3.729	3.427
5	2	-0.152	1.418	1.363	1.231	1.429	1.366	1.238	1.416	1.349	1.166	1.472	1.429	1.240
		-0.322	1.316	1.007	1.027	1.376	1.068	1.043	1.382	1.099	1.104	1.364	1.047	1.094
		-0.512	1.487	0.959	1.055	1.616	1.001	1.141	1.618	1.039	1.170	1.474	0.907	1.027
3	3	-0.152	1.897	1.992	1.688	1.842	1.942	1.700	1.842	2.015	1.759	1.847	1.966	1.655
		-0.322	1.953	1.775	1.718	1.916	1.757	1.680	1.918	1.801	1.669	1.887	1.888	1.819
		-0.512	1.991	1.677	1.651	2.083	1.752	1.731	2.015	1.839	1.780	1.896	1.782	1.759
5	5	-0.152	3.018	2.861	2.623	2.953	3.014	2.603	3.173	3.009	2.785	3.039	2.904	2.653
		-0.322	2.860	2.684	2.580	3.166	2.858	2.652	3.059	2.739	2.674	2.830	2.547	2.495
		-0.512	2.816	2.506	2.451	2.913	2.617	2.569	3.182	2.622	2.717	2.852	2.434	2.482
10	10	-0.152	3.254	3.438	3.023	3.196	3.080	2.783	3.250	3.137	2.874	2.979	3.388	2.873
		-0.322	3.201	3.165	2.653	3.057	2.959	2.777	3.372	3.183	2.907	2.717	3.039	2.830
		-0.512	3.033	2.574	2.389	2.744	2.209	2.092	3.070	2.578	2.453	2.119	2.370	2.191
Average			2.837	2.720	2.436	2.915	2.639	2.469	2.972	2.782	2.520	2.812	2.744	2.442



**Table 5. The performance of improved heuristics (n=10)**

			Heuristic algorithms											
			FL_EFSR	FL_EBSR	FL_PI	NEH_EFSR	NEH_EBSR	NEH_PI	WY_EFSR	WY_EBSR	WY_PI	RZ_EFSR	RZ_EBSR	RZ_PI
<i>l</i>	<i>m</i>	<i>a</i>	Mean Error Percentage(%)											
2	2	-0.152	2.909	2.781	2.428	3.185	2.873	2.460	3.041	2.805	2.375	3.213	2.895	2.581
		-0.322	2.818	2.488	2.235	3.024	2.538	2.414	2.966	2.484	2.398	3.029	2.574	2.484
		-0.512	2.663	1.967	1.982	2.812	1.894	2.034	2.766	1.941	2.048	2.766	2.133	2.204
3	2	-0.152	4.255	4.125	3.915	4.250	4.111	3.820	4.115	4.027	3.839	4.373	4.333	4.113
		-0.322	3.801	3.460	3.160	3.935	3.538	3.281	4.165	3.565	3.333	3.857	3.645	3.477
		-0.512	4.035	3.189	2.964	3.840	3.060	2.831	3.946	3.143	2.902	3.836	3.305	3.232
5	2	-0.152	5.479	5.167	4.649	5.444	5.384	4.697	5.581	5.454	4.879	5.746	5.744	5.219
		-0.322	5.407	5.096	4.575	5.355	5.043	4.651	5.806	5.192	4.746	5.428	5.095	4.755
		-0.512	5.359	4.515	4.445	5.507	4.431	4.343	5.853	4.760	4.480	4.874	4.098	4.013
10	2	-0.152	5.374	4.858	4.455	5.567	5.207	4.707	5.405	5.154	4.368	5.548	5.146	4.753
		-0.322	5.618	4.783	4.090	5.183	4.780	4.192	5.826	5.530	4.355	5.287	5.238	4.415
		-0.512	5.981	5.117	4.049	5.263	4.212	3.603	5.711	5.141	4.225	5.501	4.857	3.966
3	2	-0.152	2.301	2.319	1.849	2.500	2.315	1.964	2.451	2.384	1.930	2.535	2.455	2.075
		-0.322	2.217	1.978	1.696	2.253	1.983	1.772	2.283	2.005	1.819	2.325	2.089	1.909
		-0.512	2.149	1.715	1.741	2.318	1.760	1.764	2.353	1.805	1.761	2.268	1.769	1.739
3	2	-0.152	3.574	3.322	2.979	3.587	3.332	2.880	3.601	3.412	2.924	3.534	3.265	2.898
		-0.322	3.027	2.783	2.629	3.360	2.927	2.628	3.537	2.890	2.673	3.254	2.873	2.578
		-0.512	3.274	2.518	2.392	3.023	2.307	2.149	3.256	2.472	2.285	3.149	2.671	2.490
5	2	-0.152	4.576	4.426	3.822	4.942	4.736	4.054	4.950	4.590	4.070	4.859	4.547	4.192
		-0.322	4.244	4.120	3.702	4.535	4.019	3.600	4.438	4.214	3.874	4.222	3.855	3.699
		-0.512	4.275	3.649	3.363	4.394	3.573	3.506	4.501	3.863	3.841	4.190	3.824	3.591
10	2	-0.152	4.819	4.488	4.119	4.814	4.360	4.160	4.979	4.431	4.334	4.808	4.523	4.206
		-0.322	5.254	4.560	3.984	4.943	4.318	3.945	5.354	4.742	4.171	4.861	4.494	3.984
		-0.512	5.248	4.507	3.918	5.313	4.063	4.024	5.780	4.554	4.201	5.067	4.585	3.999
5	2	-0.152	1.535	1.638	1.332	1.565	1.627	1.388	1.549	1.656	1.350	1.626	1.684	1.425
		-0.322	1.435	1.350	1.245	1.667	1.451	1.290	1.648	1.443	1.302	1.552	1.414	1.342
		-0.512	1.749	1.469	1.324	1.876	1.449	1.383	1.744	1.352	1.276	1.681	1.511	1.376
3	2	-0.152	2.442	2.401	2.076	2.450	2.407	2.125	2.473	2.404	2.138	2.416	2.400	2.067
		-0.322	2.557	2.209	2.052	2.528	2.132	1.942	2.473	2.162	1.974	2.466	2.137	2.032
		-0.512	2.599	1.939	1.908	2.474	1.954	1.931	2.709	2.080	1.970	2.550	2.080	2.079
5	2	-0.152	3.515	3.294	3.115	3.529	3.363	3.072	3.530	3.401	3.112	3.518	3.296	3.024
		-0.322	3.317	2.996	2.866	3.427	3.076	2.977	3.543	3.206	3.059	3.304	3.046	2.925
		-0.512	3.318	2.998	2.831	3.621	2.789	2.717	3.636	2.775	2.761	3.117	3.065	2.919
10	2	-0.152	3.964	4.043	3.489	4.093	3.947	3.779	4.416	4.128	3.973	4.137	3.981	3.824
		-0.322	3.782	3.859	3.584	3.835	3.633	3.452	4.079	3.993	3.667	3.535	3.913	3.550
		-0.512	3.183	3.011	2.815	3.344	2.744	2.533	3.375	3.179	2.806	2.758	3.004	2.800
Average			3.668	3.309	2.993	3.715	3.259	3.001	3.828	3.398	3.089	3.644	3.376	3.109

**Table 6. The performance of four GAs ( $n=8$ )**

			Algorithms											
<i>l</i>	<i>m</i>	<i>a</i>	FL	FL_PI	FL_PI+GA	NEH	NEH_PI	NEH_PI+GA	WY	WY_PI	WY_PI+GA	RZ	RZ_PI	RZ_PI+GA
			Mean Error Percentage(%)											
2	2	-0.152	5.535	2.054	1.026	5.448	2.047	1.047	5.302	1.935	0.995	5.516	2.081	0.941
		-0.322	4.865	1.536	0.765	4.958	1.684	1.071	4.962	1.715	0.995	5.176	1.956	0.883
		-0.512	4.577	1.441	0.662	5.040	1.643	0.912	4.678	1.625	0.777	4.727	1.517	0.875
	3	-0.152	6.805	2.771	1.285	6.844	3.106	1.314	6.720	2.944	1.449	7.271	2.982	1.270
		-0.322	6.591	2.490	1.221	6.602	2.805	1.442	6.293	2.625	1.460	6.821	2.712	1.140
		-0.512	6.385	2.367	1.229	6.617	2.467	1.181	6.795	2.598	1.305	6.279	2.446	1.181
	5	-0.152	8.731	3.656	1.876	8.403	3.749	1.761	9.226	3.829	1.705	8.639	3.696	1.851
		-0.322	8.675	3.596	1.588	8.812	3.464	1.664	9.268	3.987	1.918	8.833	3.724	1.640
		-0.512	9.184	3.309	1.477	8.352	3.094	1.497	8.956	3.572	1.527	8.302	3.348	1.510
10	-0.152	7.309	3.127	1.475	7.283	3.462	1.687	7.578	3.177	1.617	7.550	3.299	1.516	
	-0.322	8.753	3.323	1.485	7.843	3.332	1.535	8.534	3.203	1.727	9.207	3.166	1.474	
	-0.512	10.015	3.150	1.380	9.209	3.299	1.355	10.235	3.219	1.279	10.637	3.218	1.167	
3	2	-0.152	4.100	1.620	0.726	4.067	1.628	0.774	4.074	1.666	0.666	3.978	1.541	0.751
		-0.322	3.761	1.436	0.642	3.790	1.569	0.687	3.978	1.599	0.630	3.677	1.331	0.655
		-0.512	3.668	1.321	0.622	3.822	1.429	0.672	3.870	1.425	0.675	3.761	1.299	0.672
	3	-0.152	5.840	2.648	1.039	5.417	2.452	1.080	5.776	2.601	1.107	5.702	2.496	1.029
		-0.322	5.484	2.360	0.885	5.174	2.269	0.883	5.443	2.277	1.067	5.278	2.021	0.847
		-0.512	5.776	2.353	0.971	5.615	2.148	0.969	5.776	2.188	1.070	5.416	2.033	0.840
	5	-0.152	6.622	3.249	1.356	6.829	3.297	1.441	6.977	3.292	1.387	6.827	3.110	1.378
		-0.322	7.247	3.242	1.329	7.518	3.237	1.344	7.278	3.094	1.292	6.926	3.049	1.303
		-0.512	7.878	2.830	1.122	7.841	3.176	1.240	8.412	3.181	1.331	7.939	3.001	1.112
10	-0.152	6.487	2.998	1.576	7.077	3.199	1.546	6.959	3.046	1.430	6.723	3.125	1.473	
	-0.322	8.839	3.306	1.190	8.426	3.291	1.311	8.995	3.517	1.304	8.598	3.233	1.322	
	-0.512	11.162	3.445	0.958	10.371	3.034	0.966	10.982	3.349	0.982	11.344	3.427	0.967	
5	2	-0.152	2.713	1.231	0.486	2.755	1.238	0.494	2.702	1.166	0.497	2.872	1.240	0.531
		-0.322	2.544	1.027	0.418	2.629	1.043	0.526	2.676	1.104	0.557	2.724	1.094	0.464
		-0.512	2.810	1.055	0.467	2.927	1.141	0.615	3.012	1.170	0.619	2.780	1.027	0.456
	3	-0.152	3.962	1.688	0.772	3.782	1.700	0.754	3.871	1.759	0.772	3.840	1.655	0.765
		-0.322	3.858	1.718	0.790	3.933	1.680	0.916	3.765	1.669	0.752	3.921	1.819	0.699
		-0.512	3.874	1.651	0.713	3.917	1.731	0.797	3.939	1.780	0.740	4.142	1.759	0.684
	5	-0.152	5.611	2.623	1.113	5.533	2.603	1.113	5.869	2.785	1.128	5.381	2.653	1.194
		-0.322	5.739	2.580	1.137	6.029	2.652	1.133	5.881	2.674	1.033	5.586	2.495	0.954
		-0.512	6.766	2.451	0.817	6.448	2.569	0.931	6.833	2.717	1.022	6.610	2.482	0.950
10	-0.152	6.112	3.023	1.394	6.035	2.783	1.432	5.955	2.874	1.384	6.382	2.873	1.332	
	-0.322	8.089	2.653	1.111	7.454	2.777	1.109	8.036	2.907	1.073	7.704	2.830	0.925	
	-0.512	9.469	2.389	0.616	8.075	2.092	0.718	9.746	2.453	0.754	8.370	2.191	0.672	
Average			6.273	2.437	1.048	6.135	2.469	1.109	6.371	2.520	1.112	6.262	2.442	1.040

**Table 7. The performance of four GAs (n=10)**

			Algorithms											
<i>l</i>	<i>m</i>	<i>a</i>	FL	FL_PI	FL_PI+GA	NEH	NEH_PI	NEH_PI+GA	WY	WY_PI	WY_PI+GA	RZ	RZ_PI	RZ_PI+GA
			Mean Error Percentage(%)											
2	2	-0.152	5.739	2.428	1.736	5.937	2.460	1.714	5.936	2.375	1.702	6.023	2.581	1.837
		-0.322	5.645	2.235	1.671	5.752	2.414	1.782	5.672	2.398	1.673	5.562	2.484	1.772
		-0.512	5.132	1.982	1.579	5.056	2.034	1.726	5.121	2.048	1.619	5.253	2.204	1.656
	3	-0.152	7.748	3.915	2.413	7.720	3.820	2.397	7.382	3.839	2.503	7.839	4.113	2.669
		-0.322	7.432	3.160	2.121	7.322	3.281	2.342	7.532	3.333	2.055	7.252	3.477	2.378
		-0.512	7.734	2.964	2.022	7.342	2.831	1.987	7.227	2.902	1.921	6.963	3.232	2.163
	5	-0.152	9.420	4.649	3.127	9.585	4.697	2.838	9.776	4.879	3.165	9.730	5.219	3.235
		-0.322	9.691	4.575	2.869	9.559	4.651	2.617	10.159	4.746	3.045	9.635	4.755	2.907
		-0.512	9.703	4.445	2.747	9.858	4.343	2.517	10.511	4.480	2.727	8.908	4.013	2.587
10	-0.152	8.587	4.455	3.129	9.182	4.707	3.150	9.314	4.368	3.093	9.196	4.753	3.129	
	-0.322	10.040	4.090	2.788	9.735	4.192	2.641	11.367	4.355	2.967	9.969	4.415	2.649	
	-0.512	11.959	4.049	2.306	10.754	3.603	2.118	12.452	4.225	2.497	11.313	3.966	2.310	
3	2	-0.152	4.330	1.849	1.294	4.610	1.964	1.276	4.644	1.930	1.262	4.609	2.075	1.241
		-0.322	4.188	1.696	1.240	4.345	1.772	1.331	4.336	1.819	1.266	4.408	1.909	1.207
		-0.512	4.029	1.741	1.327	4.333	1.764	1.265	4.343	1.761	1.243	4.137	1.739	1.093
	3	-0.152	6.119	2.979	1.878	6.211	2.880	1.768	6.285	2.924	1.814	6.074	2.898	1.834
		-0.322	5.547	2.629	1.689	6.035	2.628	1.541	6.042	2.673	1.629	5.829	2.578	1.702
		-0.512	5.779	2.392	1.542	5.456	2.149	1.484	5.717	2.285	1.363	5.824	2.490	1.493
	5	-0.152	7.886	3.822	2.400	8.269	4.054	2.389	8.021	4.070	2.355	7.881	4.192	2.516
		-0.322	7.457	3.702	2.299	7.559	3.600	2.130	7.566	3.874	2.147	7.318	3.699	2.192
		-0.512	8.060	3.363	1.891	7.885	3.506	2.003	8.274	3.841	2.249	8.096	3.591	2.085
10	-0.152	7.712	4.119	2.832	7.932	4.160	2.733	7.907	4.334	2.571	7.893	4.206	2.522	
	-0.322	9.509	3.984	2.588	8.783	3.945	2.411	9.822	4.171	2.491	8.989	3.984	2.452	
	-0.512	11.467	3.918	2.113	11.201	4.024	2.007	12.167	4.201	2.298	11.488	3.999	1.906	
5	2	-0.152	2.882	1.332	0.889	2.925	1.388	0.890	2.905	1.350	0.877	2.980	1.425	0.809
		-0.322	2.841	1.245	0.870	3.044	1.290	0.863	3.026	1.302	0.746	2.867	1.342	0.778
		-0.512	3.195	1.324	0.914	3.336	1.383	0.941	3.272	1.276	0.863	3.212	1.376	0.824
	3	-0.152	4.223	2.076	1.324	4.280	2.125	1.177	4.250	2.138	1.221	4.152	2.067	1.199
		-0.322	4.448	2.052	1.241	4.299	1.942	1.100	4.325	1.974	1.179	4.219	2.032	1.129
		-0.512	4.705	1.908	1.136	4.389	1.931	1.039	4.704	1.970	1.128	4.653	2.079	1.051
	5	-0.152	5.617	3.115	1.777	5.694	3.072	1.879	5.648	3.112	1.725	5.705	3.024	1.838
		-0.322	5.619	2.866	1.623	5.634	2.977	1.608	5.938	3.059	1.677	5.647	2.925	1.655
		-0.512	6.121	2.831	1.539	6.013	2.717	1.554	6.191	2.761	1.529	6.170	2.919	1.571
10	-0.152	6.611	3.489	2.301	6.709	3.779	2.331	6.904	3.973	2.457	6.704	3.824	2.400	
	-0.322	7.961	3.584	2.075	7.342	3.452	1.815	8.304	3.667	1.907	7.949	3.550	1.995	
	-0.512	9.352	2.815	1.250	8.223	2.533	1.065	9.911	2.806	1.167	8.865	2.800	1.364	
Average			6.791	2.994	1.904	6.731	3.002	1.845	7.026	3.089	1.893	6.759	3.109	1.893

**Table 8. The performance of four GAs ( $n=50$ )**

<i>l</i>	<i>m</i>	<i>a</i>	Algorithms							
			FL PI	FL PI+GA	NEH PI	NEH PI+GA	WY PI	WY PI+GA	RZ PI	RZ PI+GA
			RPD (%)							
2	2	-0.152	0.282	0.282 (1)	0.375	0.358 (2)	0.523	0.476 (3)	0.732	0.726 (4)
		-0.322	0.344	0.271 (1)	0.481	0.433 (2)	0.564	0.482 (3)	0.861	0.817 (4)
		-0.512	0.405	0.431 (1)	0.620	0.600 (2)	0.714	0.623 (3)	0.911	0.920 (4)
	3	-0.152	0.431	0.430 (1)	0.650	0.638 (3)	0.560	0.508 (2)	1.386	1.256 (4)
		-0.322	0.459	0.501 (1)	0.757	0.714 (3)	0.743	0.635 (2)	1.439	1.277 (4)
		-0.512	0.550	0.584 (1)	0.846	0.843 (2)	0.915	0.869 (3)	1.320	1.212 (4)
	5	-0.152	0.483	0.491 (1)	0.970	0.827 (3)	0.766	0.718 (2)	2.046	1.570 (4)
		-0.322	0.594	0.593 (1)	1.217	1.030 (3)	0.894	0.853 (2)	1.972	1.659 (4)
		-0.512	0.859	0.761 (1)	1.279	1.182 (2)	1.301	1.201 (3)	1.906	1.519 (4)
10	-0.152	0.701	0.644 (1)	1.247	1.080 (3)	1.129	0.865 (2)	2.146	1.650 (4)	
	-0.322	0.718	0.716 (1)	1.230	1.141 (2)	1.380	1.195 (3)	1.749	1.399 (4)	
	-0.512	0.955	0.913 (1)	1.444	1.349 (2)	2.062	1.621 (4)	1.746	1.518 (3)	
3	2	-0.152	0.187	0.227 (1)	0.265	0.302 (2)	0.297	0.359 (3)	0.503	0.502 (4)
		-0.322	0.227	0.236 (1)	0.370	0.391 (3)	0.363	0.355 (2)	0.535	0.541 (4)
		-0.512	0.340	0.367 (1)	0.457	0.531 (4)	0.440	0.473 (2)	0.404	0.486 (3)
	3	-0.152	0.314	0.277 (1)	0.462	0.423 (3)	0.453	0.364 (2)	0.939	0.803 (4)
		-0.322	0.336	0.377 (1)	0.590	0.601 (3)	0.553	0.494 (2)	0.972	0.894 (4)
		-0.512	0.412	0.402 (1)	0.728	0.729 (3)	0.687	0.658 (2)	0.929	0.948 (4)
	5	-0.152	0.454	0.436 (1)	0.775	0.643 (2)	0.793	0.674 (3)	0.948	0.821 (4)
		-0.322	0.692	0.670 (2)	0.650	0.625 (1)	0.959	0.827 (4)	0.806	0.714 (3)
		-0.512	0.889	0.763 (2)	0.883	0.922 (3)	1.213	1.046 (4)	0.727	0.704 (1)
10	-0.152	0.544	0.482 (1)	0.736	0.729 (2)	1.057	0.836 (3)	1.098	0.893 (4)	
	-0.322	0.814	0.750 (1)	0.917	0.794 (2)	1.190	0.962 (4)	1.039	0.886 (3)	
	-0.512	0.854	0.733 (1)	0.934	0.974 (2)	1.357	1.239 (4)	1.051	1.097 (3)	
5	2	-0.152	0.190	0.187 (1)	0.246	0.245 (3)	0.210	0.193 (2)	0.284	0.259 (4)
		-0.322	0.191	0.253 (3)	0.207	0.234 (2)	0.133	0.229 (1)	0.278	0.352 (4)
		-0.512	0.186	0.245 (2)	0.229	0.261 (3)	0.133	0.206 (1)	0.309	0.372 (4)
	3	-0.152	0.190	0.234 (1)	0.293	0.269 (2)	0.345	0.339 (3)	0.394	0.431 (4)
		-0.322	0.357	0.322 (1)	0.431	0.404 (3)	0.365	0.374 (2)	0.481	0.477 (4)
		-0.512	0.461	0.426 (2)	0.587	0.515 (3)	0.630	0.603 (4)	0.459	0.421 (1)
	5	-0.152	0.332	0.306 (1)	0.472	0.373 (3)	0.418	0.364 (2)	0.487	0.431 (4)
		-0.322	0.375	0.352 (1)	0.536	0.509 (3)	0.614	0.592 (4)	0.388	0.375 (2)
		-0.512	0.572	0.533 (2)	0.654	0.638 (3)	0.820	0.763 (4)	0.381	0.426 (1)
10	-0.152	0.503	0.446 (1)	0.563	0.568 (3)	0.536	0.557 (2)	0.698	0.573 (4)	
	-0.322	0.483	0.467 (1)	0.726	0.547 (3)	0.762	0.623 (4)	0.596	0.480 (2)	
	-0.512	0.775	0.628 (2)	0.807	0.832 (4)	1.009	0.815 (3)	0.475	0.450 (1)	
Average			0.484	0.464	0.684	0.645	0.746	0.666	0.927	0.829

**Table 9. The performance of four GAs ( $n=100$ )**

<i>l</i>	<i>m</i>	<i>a</i>	Algorithms							
			FL PI	FL PI+GA	NEH PI	NEH PI+GA	WY PI	WY PI+GA	RZ PI	RZ PI+GA
			RPD (%)							
2	2	-0.152	0.198	0.182 (1)	0.325	0.351 (2)	0.359	0.371 (3)	0.674	0.634 (4)
		-0.322	0.277	0.308 (1)	0.408	0.402 (2)	0.500	0.494 (3)	0.711	0.696 (4)
		-0.512	0.308	0.262 (1)	0.450	0.429 (2)	0.567	0.557 (3)	0.727	0.710 (4)
	3	-0.152	0.250	0.209 (1)	0.466	0.444 (2)	0.511	0.447 (3)	1.097	0.977 (4)
		-0.322	0.196	0.213 (1)	0.602	0.576 (2)	0.601	0.634 (3)	1.053	1.008 (4)
		-0.512	0.295	0.275 (1)	0.777	0.787 (3)	0.767	0.772 (2)	1.201	1.191 (4)
	5	-0.152	0.373	0.371 (1)	0.769	0.725 (3)	0.716	0.644 (2)	1.813	1.623 (4)
		-0.322	0.430	0.372 (1)	0.858	0.831 (3)	0.731	0.682 (2)	1.713	1.551 (4)
		-0.512	0.600	0.550 (1)	0.999	0.918 (2)	1.231	1.155 (3)	1.919	1.685 (4)
10	-0.152	0.407	0.364 (1)	1.005	0.838 (2)	1.030	0.877 (3)	2.320	1.811 (4)	
	-0.322	0.549	0.459 (1)	0.866	0.833 (2)	1.255	1.116 (3)	1.809	1.596 (4)	
	-0.512	0.728	0.723 (1)	1.039	1.009 (2)	1.734	1.780 (4)	1.943	1.747 (3)	
3	2	-0.152	0.133	0.138 (1)	0.186	0.199 (2)	0.232	0.266 (3)	0.397	0.382 (4)
		-0.322	0.147	0.175 (1)	0.270	0.276 (2)	0.294	0.296 (3)	0.406	0.392 (4)
		-0.512	0.225	0.202 (1)	0.348	0.339 (2)	0.441	0.438 (3)	0.465	0.441 (4)
	3	-0.152	0.265	0.247 (1)	0.365	0.330 (3)	0.280	0.294 (2)	0.623	0.551 (4)
		-0.322	0.316	0.282 (1)	0.398	0.381 (3)	0.378	0.376 (2)	0.678	0.599 (4)
		-0.512	0.346	0.375 (1)	0.652	0.676 (3)	0.565	0.532 (2)	0.781	0.791 (4)
	5	-0.152	0.368	0.307 (1)	0.486	0.456 (2)	0.473	0.458 (3)	1.101	0.931 (4)
		-0.322	0.359	0.360 (1)	0.637	0.578 (2)	0.679	0.635 (3)	0.923	0.786 (4)
		-0.512	0.531	0.513 (1)	0.851	0.807 (3)	0.973	0.973 (4)	0.861	0.784 (2)
10	-0.152	0.432	0.391 (1)	0.752	0.624 (2)	0.777	0.697 (3)	1.244	0.912 (4)	
	-0.322	0.397	0.366 (1)	0.679	0.702 (2)	0.931	0.848 (4)	0.997	0.800 (3)	
	-0.512	0.626	0.592 (1)	0.801	0.845 (3)	1.457	1.356 (4)	0.858	0.716 (2)	
5	2	-0.152	0.106	0.091 (1)	0.131	0.115 (2)	0.150	0.161 (3)	0.213	0.213 (4)
		-0.322	0.157	0.146 (1)	0.190	0.180 (2)	0.168	0.181 (3)	0.283	0.276 (4)
		-0.512	0.230	0.224 (1)	0.327	0.316 (3)	0.351	0.348 (4)	0.287	0.301 (2)
	3	-0.152	0.208	0.182 (1)	0.194	0.230 (2)	0.249	0.258 (3)	0.332	0.338 (4)
		-0.322	0.234	0.214 (1)	0.269	0.261 (2)	0.320	0.306 (3)	0.362	0.353 (4)
		-0.512	0.387	0.403 (2)	0.472	0.468 (3)	0.562	0.562 (4)	0.328	0.371 (1)
	5	-0.152	0.260	0.244 (1)	0.335	0.344 (2)	0.438	0.391 (4)	0.427	0.367 (3)
		-0.322	0.271	0.270 (1)	0.462	0.461 (3)	0.607	0.590 (4)	0.431	0.407 (2)
		-0.512	0.470	0.411 (1)	0.728	0.678 (3)	0.954	0.893 (4)	0.503	0.475 (2)
10	-0.152	0.348	0.318 (1)	0.407	0.358 (2)	0.540	0.453 (3)	0.556	0.496 (4)	
	-0.322	0.471	0.464 (2)	0.523	0.486 (3)	0.837	0.808 (4)	0.481	0.426 (1)	
	-0.512	0.681	0.551 (2)	0.844	0.813 (3)	1.276	1.125 (4)	0.448	0.334 (1)	
Average			0.349	0.326 (1.1528)	0.551	0.529 (2.5000)	0.664	0.632 (3.4028)	0.860	0.768 (2.9444)

**Table 10. Family of hypotheses ordered by  $p$ -values and other information to conduct pairwise comparisons for large size of jobs, initial  $\alpha=0.05$ .**

<i>i</i>	hypothesis	<i>z</i>	<i>p</i> -value	$\alpha_{NM}$	$\alpha_{HM}$	$\alpha_{SH}$	$APV_{NM}$	$APV_{HM}$	$APV_{SH}$
1	WGA vs. FGA	10.4554	$<10^{-16}$	0.0083	0.0083	0.0083	$<10^{-16}$	$<10^{-16}$	$<10^{-16}$
2	RGA vs. FGA	8.3253	$<10^{-16}$	0.0083	0.01	0.0167	$<10^{-16}$	$<10^{-16}$	$<10^{-16}$
3	NGA vs. FGA	6.2602	$3.8*10^{-10}$	0.0083	0.0125	0.0167	$2.3*10^{-9}$	$1.5*10^{-9}$	$1.1*10^{-9}$
4	WGA vs. NGA	4.1952	$2.7*10^{-5}$	0.0083	0.0167	0.0167	$1.6*10^{-4}$	$8.1*10^{-5}$	$8.1*10^{-5}$
5	WGA vs. RGA	2.1301	0.0331	0.0083	0.025	0.025	0.1986	0.0662	0.0662
6	RGA vs. NGA	2.0651	0.0389	0.0083	0.05	0.05	0.2334	0.0662	0.0662