

Research Article

An Improved Genetic Algorithm for Single-Machine Inverse Scheduling Problem

Jianhui Mou,¹ Xinyu Li,¹ Liang Gao,¹ Chao Lu,¹ and Guohui Zhang²

¹ *The State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China*

² *School of Management Science and Engineering, Zhengzhou Institute of Aeronautical Industry Management, Zhengzhou 450015, China*

Correspondence should be addressed to Liang Gao; gaoliang@mail.hust.edu.cn

Received 23 April 2014; Revised 4 July 2014; Accepted 7 July 2014; Published 23 July 2014

Academic Editor: Pandian Vasant

Copyright © 2014 Jianhui Mou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The goal of the scheduling is to arrange operations on suitable machines with optimal sequence for corresponding objectives. In order to meet market requirements, scheduling systems must own enough flexibility against uncertain events. These events can change production status or processing parameters, even causing the original schedule to no longer be optimal or even to be infeasible. Traditional scheduling strategies, however, cannot cope with these cases. Therefore, a new idea of scheduling called inverse scheduling has been proposed. In this paper, the inverse scheduling with weighted completion time (SMISP) is considered in a single-machine shop environment. In this paper, an improved genetic algorithm (IGA) with a local searching strategy is proposed. To improve the performance of IGA, efficient encoding scheme, fitness evaluation mechanism, feasible initialization methods, and a local search procedure have been employed in the paper. Because of the local improving method, the proposed IGA can balance its exploration ability and exploitation ability. We adopt 27 instances to verify the effectiveness of the proposed algorithm. The experimental results illustrated that the proposed algorithm can generate satisfactory solutions. This approach also has been applied to solve the scheduling problem in the real Chinese shipyard and can bring some benefits.

1. Introduction

Usually, in previous research works, the scheduling problem is assumed to be an ideal processing condition and the processing parameters are deterministic. Because the real systems operate in highly dynamic and uncertain environments, scheduling environment is very complex. In view of this, the predictive optimal schedule may become neither feasible nor optimal. In previous scheduling policy, a rescheduling method with a new schedule sequence from changing the planned schedule has been widely used in such system. However, the traditional static scheduling procedure is suitable for such kind of problem which assumes that information of all parts and the status of job shop environment are specified. This implied that the traditional scheduling method could not solve that problem. To overcome these problems, there

is an increasingly need for the research of new scheduling method. Therefore, in this paper, a novel developing idea of scheduling “inverse scheduling problem” (ISP) is introduced to improve the existing manufacturing systems. The main difference between the mentioned scheduling problems is that the inverse scheduling assumes a prespecific job sequence and expects to turn that sequence into an optimal one by minimally adjusting the job parameters [1].

The definition of inverse scheduling problem is that the exact values of parameters (e.g., processing times and due dates) are controllable and feasible job sequences are given but optimal and prespecified job sequences do not become optimal through adjusting processing parameters for a target [2]. And Brucker and Shakhlevich [3] invented the concept of ISP, where they considered the inverse scheduling problem with maximum lateness objective. But, until recently, there

are no systematic research results except 3 published works. Furthermore, ISP is also acknowledged as one of the NP-hard [3] problems. The reason that the inverse scheduling problem has not been considered previously is in fact not due to the lack of its practical application, but due to its more complexity [4, 5]. In fact, there might be several application scenarios for ISP. Due to the limited space, we just take an example to show the application. In the modern manufacturing system, this may happen as follows, for example, when the producer arranges the job sequence only based on estimated parameters value at the beginning of production planning. But the fact is that the real value of parameters may be slightly different from the estimates. It may make the given sequences no longer optimal [6, 7]. Of course, through reality in some scheduling shop, the preplanned job sequence cannot be changed randomly due to technological or process constraints. Then, the producer may identify a few jobs that can change processing times, speeding up some of them by adding additional resources or slowing down others, so that the original sequences become optimal [8]. We could also use the same way to adjust the weights of jobs [9]. Adjusted processing parameters must guarantee that the given sequence is the best possible one for the producer. In that circumstance, processing parameters are the decision variables. Therefore, ISP also will be used in some problems with controllable processing parameters.

Recently, inverse optimization problem (IOP) [10–12] has been a hot research topic which leads to more relative relationship between the ISP and IOP (Wang 2009) [13–15]. Therefore, we pay more attention to inverse optimization (IOP) and its wide applications. By comparing the inverse optimization problems, the challenging research question is how we generate a feasible scheduling sequence at the beginning or how we obtain the optimal solution by adjusting the processing parameters? Usually, only job sequence is required to determine in traditional scheduling. However, the minimal perturbation to the job parameters is also needed to be identified in ISP. It is more difficult to assign resources in given job sequence constraints, so ISP is more complex than problems with determined parameters.

In fact, ISP methods are also a challenge research topic because almost no scholars use metaheuristic approach for the inverse scheduling problems. In addition, most research works just considered the single-machine inverse scheduling; very few papers paid attention to multiple machines. However the background of certain applications is not enough to study. To overcome these problems, there is an increasing need for deeper research on application of inverse scheduling. The ISP introduces significant improvements to the efficiency of manufacturing through eliminating or reducing scheduling conflicts. More importantly, ISP may promote new ideas and developments in shop scheduling problem.

In this research, the main contributions of this paper can be summarized as

- (1) being the first ever research to formulate an inverse scheduling problem mathematically;
- (2) introducing controllable processing parameters to the traditional single-machine scheduling problem to

consider a more realistic situation for most of the manufacturers;

- (3) combining the local search approach with the improved genetic algorithm to develop a metaheuristic algorithm for solving SMISP for the first time;
- (4) assessing the quality of the proposed algorithm and comparing it with the traditional algorithm and finally showing that 22 preferred solutions could be found among 27 experiment problems;
- (5) applying the proposed approach to solve a real shipbuilding production factory and showing that a saving of 12% is obtainable by the proposed approach.

The remainder of this paper is organized as follows. Section 2 is the literature review on single-machine scheduling problem and ISP. Section 3 is the problem formulation. A novel IGA algorithm for SMISP is proposed in Section 4. Experimental comparisons and case studies are reported in Sections 5 and 6. Section 7 is conclusions and future researches.

2. Literature Review

2.1. Brief Review of Inverse Scheduling Problem. There are only a very few papers that have investigated the inverse scheduling. Brucker and Shakhlevich [3] firstly raised a particular definition of the inverse scheduling. This paper studied two novel inverse scheduling problems, namely, the single-machine inverse and reverse scheduling problem, respectively. The two problems with adjustable due dates or processing times are proving NP-hard; for the remaining problems, they provided solution methods for the corresponding mathematical programming formulations. Unfortunately, there is no effective method; only linear programming methods are used for SMISP in previous literature. One year later, Brucker addressed a more complicated two-machine flowshop inverse scheduling problem. They verified the necessary and sufficient conditions of optimality of a solution which was obtained in flowshop ISP [2]. An earlier research performed by Koulamas [1] has shown that such inverse problem can be simplified linear programming problem even though the corresponding forward problem cannot be solved in polynomial time. However, he does not consider possible increases in the scheduling criteria.

And with the development of inverse optimization, inverse scheduling already caused the attention of some domestic scholars. Typically, Chen et al. [16] proposed the mathematical programming tool for a single-machine inverse scheduling problem; the corresponding objective is to minimize the total weighted completion time and the corresponding optimal solutions are obtained under different norms. More recently, several studies focusing on the single-machine supply chain inverse problem and flowshop scheduling with two machines have been provided by Chen and Tang [17]. It was shown that, by the theories of scheduling, these problems can be formulated as respective mathematical programming involving different controllable parameters [17]. In 2012, Pham and Lu [18] also provided efficient methods

for the inverse scheduling problem with the total weighted completion time objective on identical parallel machines. It is important to note that all of the previous works only focused on developing a linear programming to find a solution for their proposed problem. To the best of our knowledge, this paper is the first to propose an improved genetic algorithm for SMISP.

Single-machine scheduling is important in shop scheduling system because it is a popular case of many industrial and services environments. The numerical results indicated that there have been many research works and have some result in single-machine problem (e.g., [19–22]). Therefore, the results of single-machine research not only provide insights into the single-machine environment but also provide a basis for intelligent algorithm for SMISP [23–30]. In order to solve the large-scale problem, it should be effective to apply the intelligent algorithm [9, 31] for SMISP to find the optimal solution. The most common intelligent algorithms applied in scheduling problems are the GA. For example, see [32–36]. In the open literature, many researchers have applied GAs to single-machine scheduling problems. GA has an efficient exploration ability to search a broader space. However, it does not have enough ability to do local search for a good solution. So, many GAs incorporate local search algorithms that enhance the performances of the algorithm. As can be seen from this review, little work has already been done in the area of inverse scheduling. Nevertheless there is still significant room for improvements in this area.

2.2. Limitations of Current Researches. Most of the above-mentioned works use only linear programming algorithm to deal with an ISP. From the scheduling model, it can be observed that a lot of research works just consider makespan minimization, while other common criteria, such as cost or tardiness, are less studied. With respect to solution approaches, linear programming and mixed integer programming are the most frequently exact procedures. Therefore, there is still a challenge in the research about ISP.

- (1) The first issue is that most research work on ISP just considers one single objective and small-scale problem. This is probably due to the fact that the single-machine problem is easier to treat than the multiple machine problems. However, most real-life situations are very complex.
- (2) Furthermore, most early papers mainly used exact solution like linear programming and so on; the limitations of such algorithm is feasible only for some simple, small-scale problems or single objective problems. But it still requires a large amount of the intelligent algorithm for solving large-sized problems or more complex inverse scheduling problem. Therefore, there seems to be a trend to propose more effective metaheuristic for ISP.

3. Problem Formulation

Unlike the forward scheduling problem, the inverse scheduling problem assumes that a given feasible solution (schedule sequence) is prespecified and the job parameters are controllable; the objective is to determine the minimal perturbation (e.g., mainly processing times or their weights) so that the original job sequence becomes an optimal solution (schedule). In this paper, an inverse scheduling problem is considered under a single-machine environment with the sum of weighted completion time. Thus, the optimization objective of the SMISP is to minimize the adjustment of the process parameters ($\|p - \bar{p}\| \|w - \bar{w}\|$). In this paper, it is assumed that weight of each job is controllable and needs to be minimally adjusted so that a prespecific schedule becomes an optimal one. Formally, w_j ($j = 1, 2, \dots, n$) are the decision variables. It is important to note that the resulting total weighted completion time based on the adjusted parameters is no greater than the total weighted completion time based on the original parameters. In solving this problem, several constraints and assumptions are made as follows.

- (i) Each machine could process at most one job at a time.
- (ii) All jobs are available at time 0.
- (iii) There are no precedence constraints among the different jobs.
- (iv) The machines are always available at zero and never break down.
- (v) Processing time and weight of all jobs are known.
- (vi) Weight of each job is adjusted and processing time is determined.
- (vii) The given job sequence is fixed.
- (viii) The given job sequence is a feasible schedule not the most optimal one.

Obviously, the optimal solution for the scheduling problem $1 \parallel \sum w_j C_j$ is obtained by scheduling the jobs according to the weighted shortest processing time (WSPT) rule [37]; that is, the optimal schedule must satisfy the condition: $p_1/w_1 \leq p_2/w_2 \leq \dots \leq p_n/w_n$. Moreover, $\sum_{j=1}^n w_j c_j = \sum_{j=1}^n p_j (w_j + \dots + w_n)$.

Therefore, SMISP problem can be formulated as follows:

$$\min Z = \min \sum \|w_j - \bar{w}_j\|, \tag{1}$$

$$\frac{\bar{w}_1}{p_1} \geq \frac{\bar{w}_2}{p_2} \geq \dots \geq \frac{\bar{w}_n}{p_n},$$

$$\sum_{j=1}^n \bar{w}_j (p_1 + \dots + p_n) \leq \sum_{j=1}^n w_j c_j, \tag{2}$$

$$\bar{w}_j \geq 0, \quad j = 1, 2, \dots, n, \tag{3}$$

where the parameters can be described as two sets; the original processing time, weight, and completion time of each job are denoted by p_j , w_j , and c_j , respectively; the adjusted

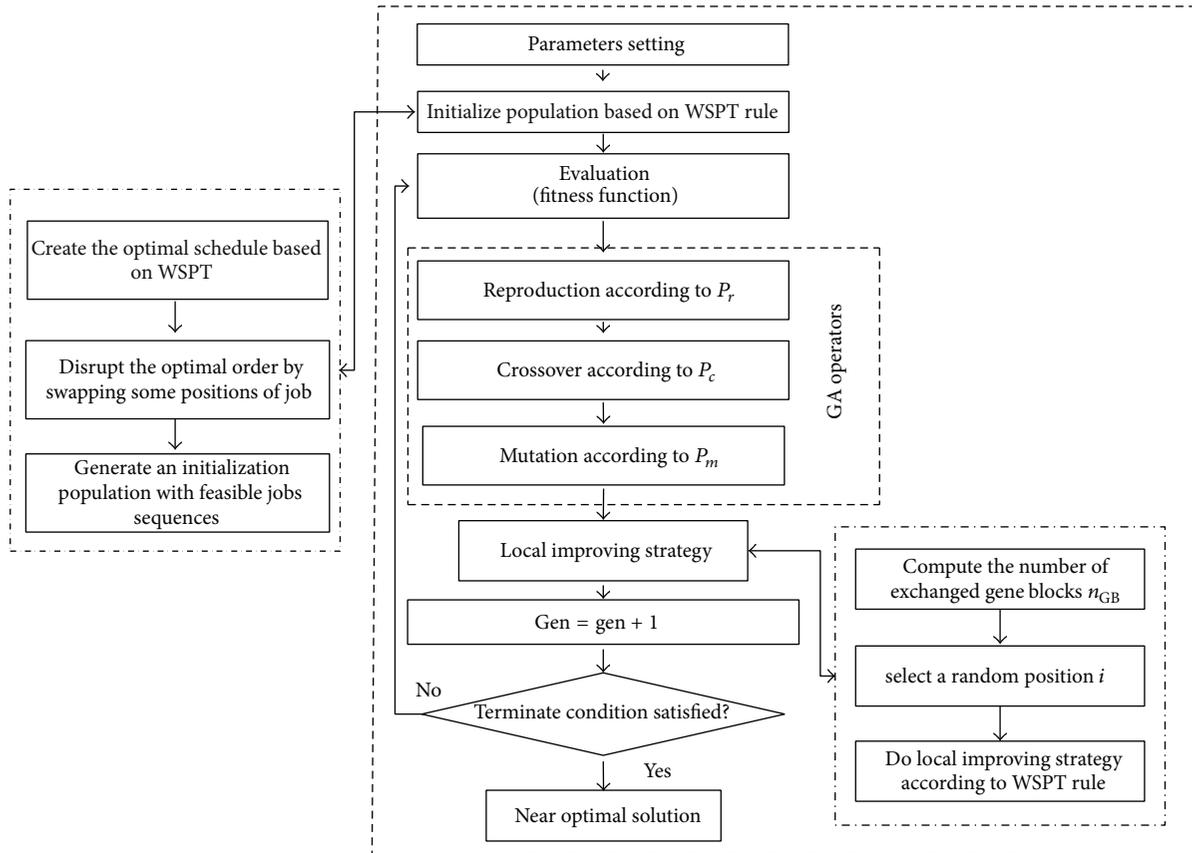


FIGURE 1: The workflow of modified IGA.

processing time, weight, and completion time of each job are denoted by \bar{p}_j , \bar{w}_j , and \bar{c}_j , respectively. Constraints (1) guarantee that job sequence is optimal under \bar{w}_j . Constraints (2) ensure that the resulting total weighted completion time based on the adjusted parameters is not greater than the total weighted completion time based on the original parameters. Constraints (3) mean that the weights can be nonzero.

4. Single-Machine Inverse Scheduling with Improved Genetic Algorithm

4.1. IGA for Single-Machine Inverse Scheduling Problem. In this research, an improved genetic algorithm is presented to solve the SMISP problem in small, medium, and large size. To improve the optimized performance of the approach, a local search procedure is integrated in IGA. Efficient encoding scheme, fitness evaluation mechanism, feasible initialization methods, and WSPT rule as local search procedure have been developed to improve the performance of IGA.

4.1.1. The Modified IGA Framework. The workflow of the proposed modified IGA is shown in Figure 1 and detailed description of the method will be given in next part. Its framework can be described as follows.

Step 1. Select the parameters of GA, such as the size of the population (P size), maximum generations ($\max G$), reproduction probability (pr), crossover probability (pc), and mutation probability (pm).

Step 2. Create the optimal schedule sequence based on WSPT rule and then disrupt the optimal order by swapping some positions of jobs; in this way can we generate an initialization population with a feasible job sequence.

Step 3. Use a proposed fitness evaluation technique to evaluate the fitness of each individual in the population and retain the best results of M size.

Step 4. Reproduce the individual directly to the next generation according to reproduction probability.

Step 5. Cross over the two-father generation individual parts of the structure to replace restructured part and generate new individual operation and then mutation operation is performed.

Step 6. Do local search for the individual based on WSPT rule by computing parameter values n_{GB} , N_M .

Step 7. Is the terminate criteria satisfied? If yes, go to Step 8; else go to Step 3.

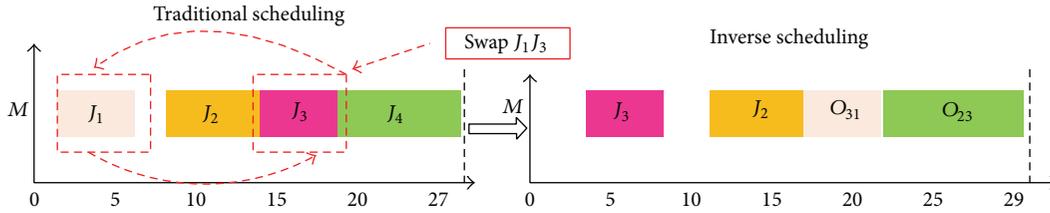


FIGURE 2: Inverse scheduling with the single machine.

Step 8. Stop the algorithm and output the near optimal solutions.

4.1.2. *Chromosome Representation and Decoding.* The weight-based representation is adopted as the encoding method because it can ensure that each value of processing parameter can be adjusted within a given limit. Each chromosome is presented by binary codes of weight. Each string contains the number of genes which is equal to the integral multiple of the job. The following example shows five jobs processed on one machine randomly. In this example, it is assumed that the job weight set = $\{W_2, W_1, W_4, W_3, W_5\}$, and the corresponding weight values = $\{13, 30, 100, 47, 15\}$. Thus, each chromosome can be interpreted as $\{0001101, 0011110, 0101111, 1100100, 0001111\}$. Decoding could obtain a determined weight value from a chromosome by scanning the gene from left to right. Not all weights require calculating again so that the decoding approach may decrease the search space and runtime.

4.2. Initialization and Fitness Evaluation

4.2.1. *Initial Population.* A new initialization method (called feasible initialization) is proposed to initialize the population at the beginning of inverse scheduling. Feasible initialization is used to generate a nonoptimal schedule sequence before inverse scheduling. This initialization method is demonstrated in Figure 2. In the single-machine inverse scheduling problem, prespecified job sequences are needed in advance. It means that a feasible scheduling sequence should be generated before inverse scheduling. It is well known that the optimal solution for the single-machine scheduling problem is obtained by the weighted shortest processing time (WSPT) rule [37]; the optimal scheduling must satisfy the decreasing order of w_i/p_i . Firstly, the optimal schedule sequence is obtained according to WSPT rule. Then, in the sequence, some points of the jobs are randomly swapped and other job positions remained. In Gas [38–41], a set of chromosomes forms a population.

4.2.2. *Fitness Evaluation Mechanism.* The purpose of the fitness evaluation is to calculate the goodness of the candidate solutions in the population with respect to the objective function. According to the objective function, fitness of chromosomes should be obtained by calculating the perturbation of job parameters. Chromosomes are decoded and the concerned objectives are obtained. For a scheduling, the fitness has been used as the objective. However,

ISP has its specific problem characteristics, so that various objectives should be considered. Firstly, this paper considers the minimal perturbation of parameters as an objective. Secondly, we must ensure that the given schedule sequence cannot be changed. Thirdly, as producers, they also want to consider possible increases in the scheduling criteria. Therefore, fitness evaluation mechanism is developed to balance the conflict. In order to perfect this method, a fitness evaluation mechanism is introduced to analyze SMISP in this paper. Evaluate two chromosomes marked (Ch1 and Ch2) as follows.

Step 1. If two chromosomes both satisfy the prespecified job sequences, then evaluate the unfit quality of the individual fitness ($\|w - \bar{w}\|$) and select a smaller one.

Step 2. If neither satisfies the prespecified job sequences, then count the number of genes which satisfies the original job order and select the larger one. For example, if $Ch1 = Ch2$, go to the next step to evaluate the fitness of individual.

Step 3. If only one chromosome satisfies the prespecified job sequences, If Ch1 maintains the pre-specified job sequences, and then select Ch1. Else select Ch2, then select Ch1 or else Ch2.

Fitness evaluation mechanism of SMISP is shown in Figure 3.

4.3. *Crossover Operator.* GA has two evolutionary operators, crossover and mutation. These operators enhance the performance of solutions by propagating similarities and unexpected genetic characteristics to offspring. This paper borrows crossover operator from the partial schedule exchange crossover, which is proposed by Ghrayeb and Damodaran [42]. This crossover procedure is demonstrated in Figure 4. Another crossover operator called one-point crossover is also adopted. The modified partial crossover operator can be summarized as follows.

Step 1. Decide a set of partial schedules and the corresponding position; it means that the genes between blocks are swapped from the two parents to the corresponding offspring.

Step 2. Pick up a starting point from parent P1 at random. The position of ending point is equal to the first position plus the length of the block. So, partial schedule 1 can be created such as the red blocks.

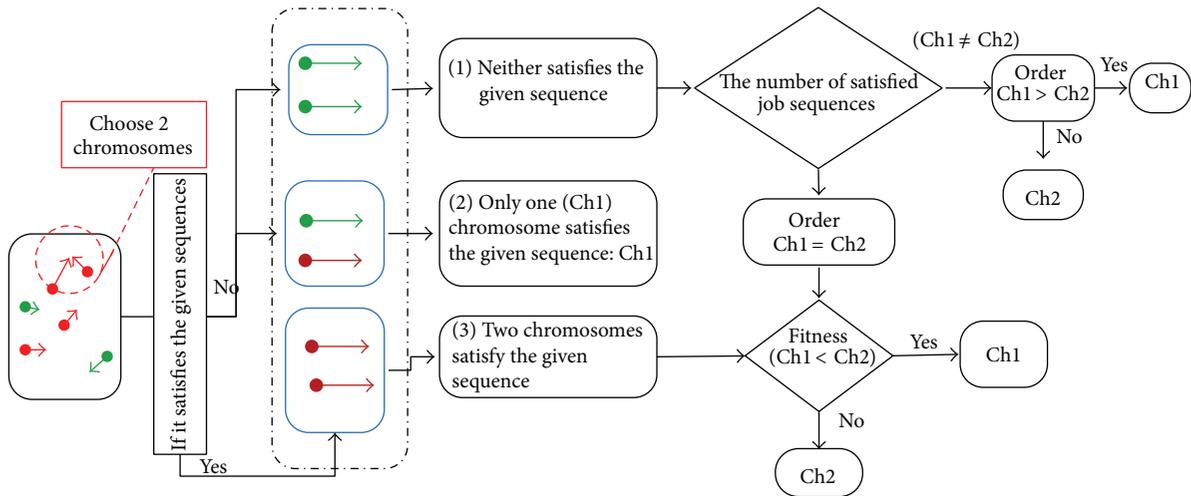


FIGURE 3: Fitness evaluation mechanism of SMISP.

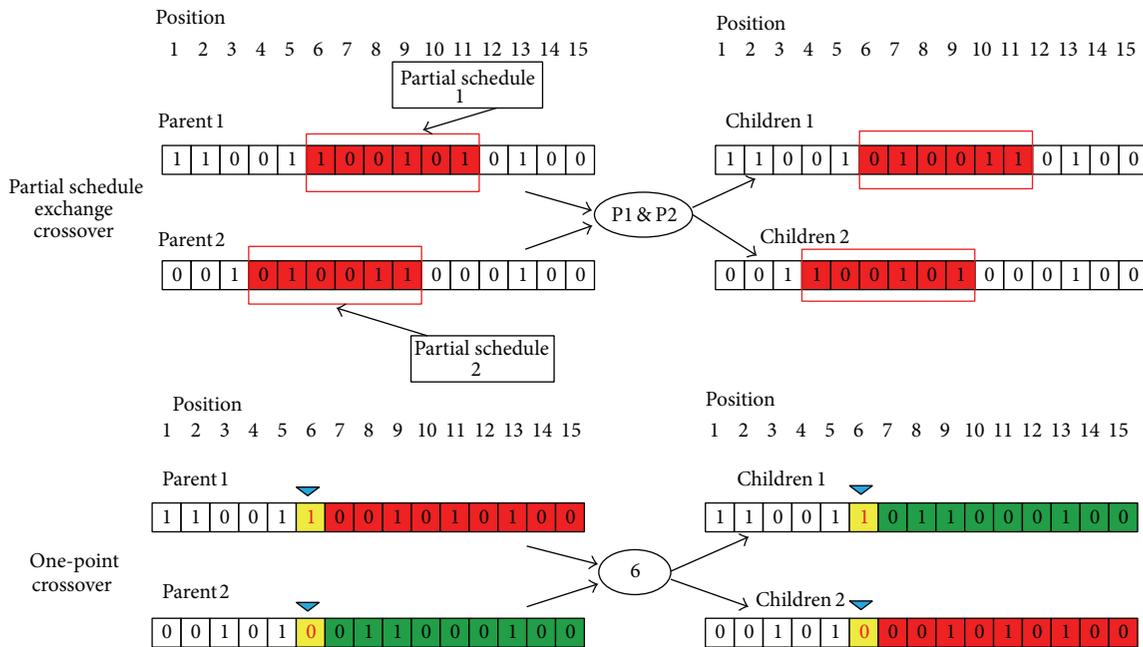


FIGURE 4: Crossover operator.

Step 3. Partial schedule 2 is as the same as Step 2.

Step 4. Form children (or offspring) C1 and C2 by swapping the block gene; for example, see Figure 4.

4.4. *Mutation Operator.* Mutation is used to produce perturbations on chromosomes to maintain the diversity of population. A familiar mutation, called swap mutation, is used in this paper. This mutation procedure is demonstrated in Figure 5.

4.5. *Selection Strategies.* The Roulette wheel [39] selection procedure is used for selecting parent individual in crossover

operator. However, a larger probability is likely to be given the better individuals due to selection strategy. Each individual will carry the possibility of selection in offspring and the selection is used to maintain diversity. The parents are selected randomly, after sorting individuals according to their objective and assigning the selection probability.

4.6. *Local Search Strategy.* GA and local search are integrated in IGA to obtain the advantages of algorithms, the exploration ability of GA, and the exploitation ability of local search. The neighborhood search can obtain new neighbor solutions by perturbation in neighboring solutions spread. In our algorithm, we use WSPT rule as acceptance strategy to

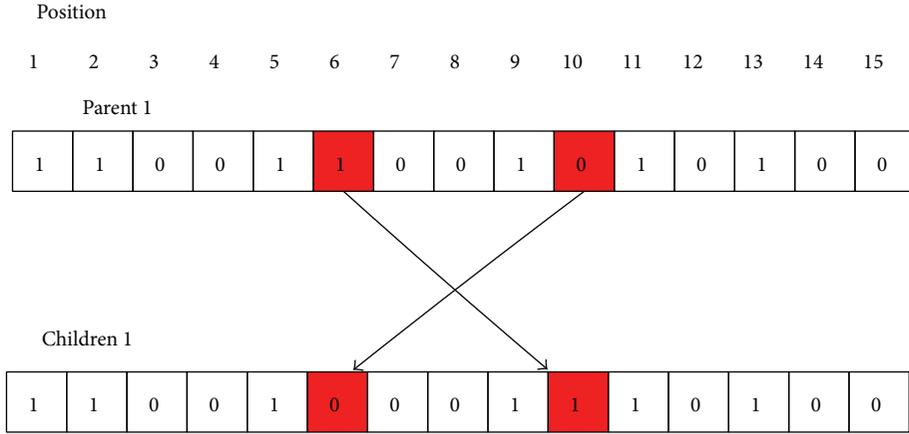


FIGURE 5: Mutation operator.

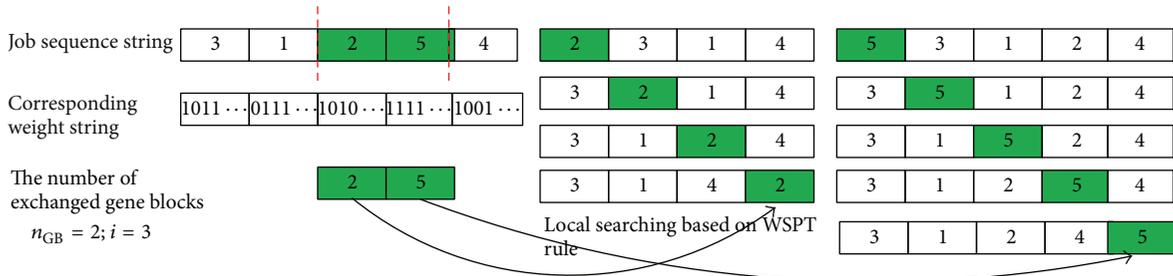


FIGURE 6: An example of applying WSPT-based local search to an individual.

decide the best partial sequence. Noticing the exploitation ability of local search and exploration ability of GA, we integrate local search and GA more tightly to perfect the algorithm performance.

Besides, reconstructing chromosome may devote much more time and decrease the efficiency of our algorithm. Thus, according to modeled NEH-based local search procedure in Chiang et al. [43], not all of the chromosomes will be converted. The improvement part is designed according to the two key parameters such as the length of exchanged gene block n_{GB} and the set of jobs positions i to m . Obviously, it is a good method to make local search (using a large neighborhood) at the early stage and to do perturbation (using a small neighborhood) at the end of the generation. So, the number of exchanged gene blocks can be denoted as $n_{GB} = \max \{N_M \times (1 - t/TG), 1\}$; this is controlled by the parameter N_M (the maximum number of exchanged jobs) and the current generation number t ; besides, the set of gene positions first selects a random position i . Then, the ending point of gene block is equal to i plus n_{GB} . An example is given in Figure 6.

5. Experimental Studies and Discussions

Some experiments have been conducted to measure the adaptability and the superiority of the proposed IGA

approach, which is a popular heuristic algorithm. The performance of the approach is satisfactory from the experiments and comparison.

5.1. Design of Experiments. The proposed IGA algorithm is coded in C++ on a PC with Pentium IV 2.29 GHz processor and 2.86 GB memory (Algorithm 1). To verify the feasibility and performance of the IGA algorithm, 27 cases have been conducted. In this section, a series of training sets and test sets are generated by the instance generation approach used by Jakobovic and Budin [44]. Each scheduling problem instance is defined by the following parameters:

- (i) n the number of jobs; tts value is 20, 50, or 100;
- (ii) p_j processing times of jobs j ; the values of processing time are assumed as integers and drawn out of $U[5, 25]$, $U[30, 100]$, or $U[80, 120]$, where U refers to the uniform distribution;
- (iii) w_j weighted values of jobs j ; the weighted values are assumed as integers and drawn out of $U[10, 50]$, $U[10, 100]$, or $U[100, 200]$;
- (iv) P size, population size: 100;
- (v) max G , stopping criterion: 500;
- (vi) pr , rate of elite chromosomes: 8%;
- (vii) pc , crossover probability: 90%;
- (viii) pm , mutation probability: 50%;

```

if (oner1.sorted > oner2.sorted) then oner1
else if (oner1.sorted == oner2.sorted)
  If (oner1.fitness < oner2.fitness) then oner1
  else if (oner1.fitness == oner2.fitness)
    {
      cw1 += (float)oner1.Wi[i] * Ci[i];
      cw2 += (float)oner2.Wi[i] * Ci[i];
    }
  if (cw1 < cw2) then cw1

```

ALGORITHM 1: The code of the fitness evaluation mechanism.

TABLE 1: Test results of SMISP.

Scenario number	Number of jobs	RPD		PRCS		The adjustment of parameter values		Computational time (ms)	
		GA	IGA	GA	IGA	GA	IGA	GA	IGA
1	20	62.4%	53.7%	6.5%	10.6%	18	13	8470	7859
2	50	48.8%	41.1%	4.96%	5.3%	44	40	15450	10419
3	100	21.1%	10.0%	3.25%	4.28%	86	83	51661	45638
4	20	35.5%	30.9%	6.4%	10.6%	19	12	9907	9005
5	50	31.4%	27.6%	4.1%	14.3%	43	41	16275	14075
6	100	33.6%	20.5%	2.2%	3.2%	84	64	24045	22090
7	20	73.3%	65.1%	4.7%	5.1%	17	12	9512	8512
8	50	48.4%	39.6%	1.5%	7.8%	42	40	14412	13902
9	100	29%	12.8%	1.5%	10.9%	85	78	24418	23398
10	20	44.3%	34.7%	1.3%	2.7%	18	13	9320	8464
11	50	46.4%	41.9%	1.3%	9.1%	44	31	15226	14027
12	100	13.9%	9.8%	-1.7%	2.2%	84	83	9668	8874
13	20	72.7%	69.5%	0.9%	2.1%	19	13	9515	8627
14	50	47.4%	37.1%	-0.9%	1.1%	44	35	26928	25095
15	100	32.1%	12.8%	-6.1%	4.9%	84	68	23578	22639
16	20	57.2%	46.2%	0.4%	3.7%	19	14	8667	7706
17	50	42.1%	36.6%	0.09%	1.5%	44	40	12904	12845
18	100	15.7%	12.6%	1.3%	1.6%	84	76	38438	37564
19	20	81%	72.7%	1.2%	2.8%	19	12	8297	7084
20	50	26.5%	18.9%	2.8%	3.6%	44	43	12715	11915
21	100	22.1%	10.1%	3.1%	9.4%	83	79	19301	18357
22	20	63.8%	55.7%	1.4%	3.9%	18	15	8290	7843
23	50	22.2%	20.1%	-1.1%	2.9%	42	32	23977	22757
24	100	17.4%	14.5%	0.9%	3.05%	83	75	20473	18809
25	20	26.4%	21.8%	6.0%	7.9%	19	15	9437	8438
26	50	38.6%	29.6%	-1.7%	0.2%	44	42	14425	13405
27	100	14.6%	12.7%	2.2%	3.9%	85	67	22931	21893
Average		39.6%	31.8%	1.72%	5.13%	48.7	42.07	17342.2	15971.8

5.2. *Computational Results and Discussions.* The literature review shows no intelligent algorithm to produce a set of trade-off solutions for the SMISP. It is worth noticing that there are some parameters to be determined in the proposed algorithm. However, the performance of the algorithm varies greatly for solving different instances, when some parameters

are set differently. We select the best parameter setting by testing (Table 2). Due to the limitation of space, we do not present the detailed testing processing. And, in the running process, if the best solution is not updated over 20 times, then the algorithm is stopped. The numerical results are compared with those reported studies using the previous

TABLE 2: Case parameter setting.

N	Project ID	Number	Production name	p_j (S)	w_j (RMB)
Center lathe C620					
1	C278J	3637235	06-6B-CB*56-1983	87	22
2	A288J	3004151	01-6B-CB*821-1984	87	14
3	F103701J	3635876	WS104.5400021G	195	143
4	A307K	3639177	20#steel/ φ 114*11/L = 150 mm	156	4
5	A308K	3639171	20#steel/ φ 114*12/L = 150 mm	111	3
6	A307K	9132344	20#steel/ φ 114*11/L = 150 mm	156	4
7	A308K	9132338	20#steel/ φ 114*12/L = 150 mm	111	3
8	A317J	3637953	08-6B-CB*56-1983	20	5
9	A317J	3637948	06-6B-CB*56-1983	780	15
10	A317J	3637946	01-6B-CB*821-1984	195	5
11	A317J	3637947	03-6B-CB*821-1984	195	7
12	A317J	3637960	01-6B-CB*56-1984	130	5
13	A317J	3637961	03-6B-CB*56-1984	130	8
14	A317J	3005568	WS104.5400021G	390	15
15	A317J	3005568	WS104.5400022G	390	23
16	A317J	3005563	WS104.5400016G	260	28
17	A317J	3005564	WS104.5400017G	390	34
18	A317J	3005565	WS104.5400018G	780	45
19	A317J	3005566	WS104.5400019G	390	41
20	A317J	3005535	WS104.5400039G	390	49

GA approaches. Performance comparison could be made between IGA and conventional algorithms.

5.2.1. Comparison of the Conventional GA and IGA Algorithms. The results of the experiments are summarized in Table 1. In order to verify the efficiency of the method, two criteria were considered here. The relative percent deviation (RPD) and the relative percentage of cost savings (RPCS) are formulated as expressions (4) and (5), respectively:

$$\text{RPD} = \frac{\text{AS} - \text{BS}}{\text{AS}} \times 100\%, \quad (4)$$

$$\text{RPCS} = \frac{\text{OBJ}_j - H_j}{\text{OBJ}_j} \times 100\%, \quad (5)$$

where AS and BS are the average value and the best value of each instance; OBJ_j is the original objective ($\sum w_j C_j$) value before adjusting the weights and H_j presents the optimal weighted completion time after adjusting. The objective function is to minimize the sum of weighted completion time. The job weights (w_j) are set by the scholar simply to reflect the relative importance of the jobs. Yet, as far as we know, for shipbuilding industry, it could also be considered as the cost or investigation attributed to processing jobs. So RPCS can be used to evaluate the percentage of cost savings; obviously, the larger the value is, the better the RPCS is. In this table, the fifth column is the minimal adjustment of each instance. Average computational time of each instance is summarized in the last column.

These results indicate that the IGA algorithm performs significantly better than the conventional genetic algorithm,

since the IGA algorithm not only produces the substantial better overall mean AS, BS, and WS in terms of the performance measures RPD\PRCS but generates the much better AS, BS, and WS for every instance as well. For each instance, the IGA algorithm can obtain much lower RPD value than the conventional algorithm. As shown in Table 1, the *average* results from IGA and GA algorithms in RPD\PRCS and run time are 39.6%, 1.72%, 17342.2, 31.8%, 5.13%, and 15971.8, respectively. Particularly, for nine large-sized instances (100 jobs), the RPD produced by the IGA algorithm is 50 percent lower, which suggests that the IGA algorithm can nearly find all the better solutions for each instance. Further analysis form PRCS shows that 5 negative solutions could be obtained among 27 experiment problems by GA. It implies an objective value greater than the original objective value; it violates our imposed constraint and it is rejected. However, IGA algorithm produces significant improvement over the 5 nonideal solutions; more importantly, 5 PRCS values achieved by the IGA algorithm are about 2.0 times higher than those by the conventional algorithm. These results indicate that IGA offers better performance for the single-machine inverse scheduling problem.

To check whether the observed differences from Table 1 are indeed statistically significant, a statistical ANOVA was carried out for analysis, where the number of jobs n , metric RPD, PRCS, and the type of algorithm are considered as factors. In this study, effects are considered significant if the P value is less than 0.05. Due to the limitation of space, we only provide the ANOVA test results for medium and large-scale instances (50 jobs and 100 jobs). Each plotted average corresponds to the average of 150 results of each instance

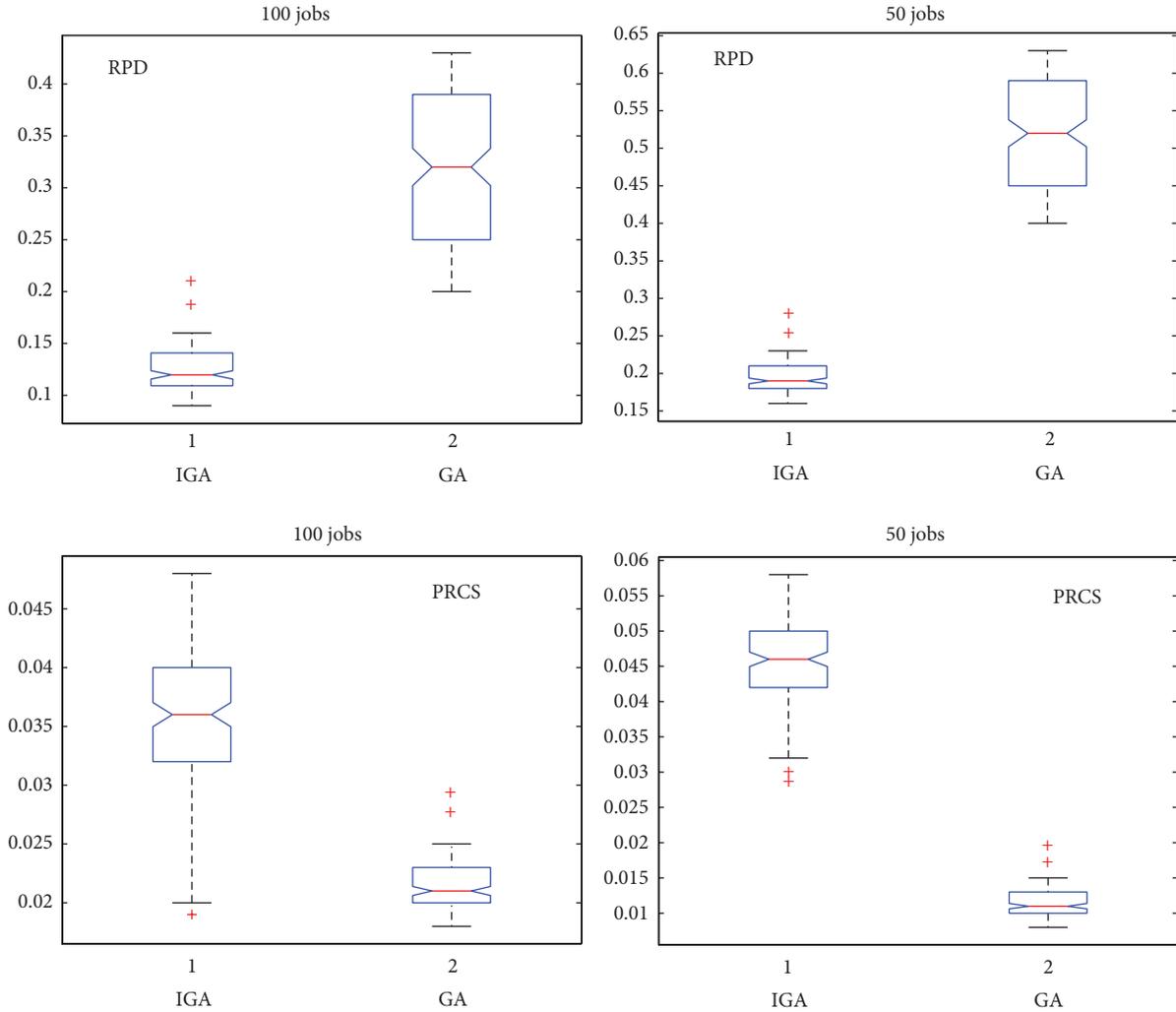


FIGURE 7: The ANOVA results for two-way analysis.

(600 results). The results are given in Figure 7. It can be seen that the metric RPDs obtained by our proposed algorithm are significantly better than those of GA. In addition, RPCS can be used to evaluate the percentage of cost savings; obviously, the larger the value is the better the RPCS is. Therefore, the ANOVA results indicate statistically significant differences in the metric PRCS at a 95% confidence level; the experimental results show that IGA can produce a set of good solutions for single-machine inverse scheduling.

5.2.2. Computational Results of IGA Algorithms. For testing the effectiveness of algorithm in small, medium, and large problems, Figure 8 shows convergence curves of objective for different series and sizes, the x -axis presents the generations, and y -axis presents the perturbation to the job weights. For example, the algorithm converges to a good value within the first 180 iterations and then no further improvements are made.

The objective of this paper is to study the minimum required perturbation in the aforementioned job weights so that each change value of processing parameter (weights) is shown in Figure 9, where blue bar graph is the original weights and while red (yellow) bar graph presents the adjusted weights. From these graphical representations, it is very clearly seen that 15 weights are decreased whereas the other 5 are increased. Note here that the adjustment trend is likely to fit with an actual production environment.

The results described in Figure 10 are simple in order to illustrate how to improve the manufacturing system by ISP. Comparisons have also been made and the relative percent deviations of the proposed algorithms for different instance have been shown in Figure 10. The x -axis presents the test number and y -axis presents the relative percent deviation explained in Section 5.2.1. These points above zero-line of X -axis show that the solutions obtained by IGA are superior to other solutions obtained by GA algorithm. From Figure 10, we see that most values are above zero-line, so the modified

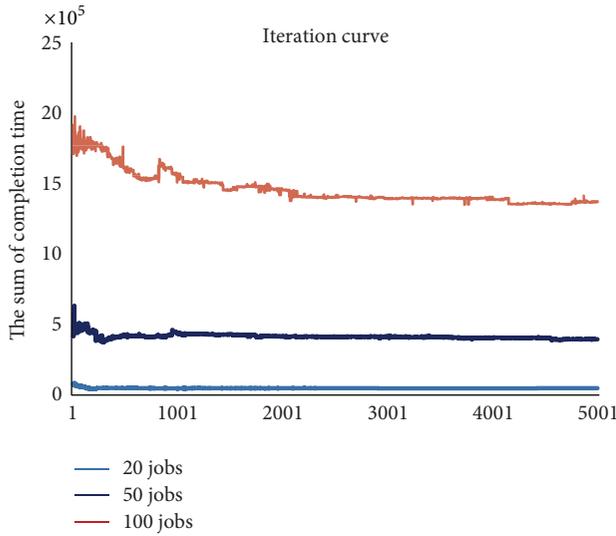


FIGURE 8: Convergence curve.

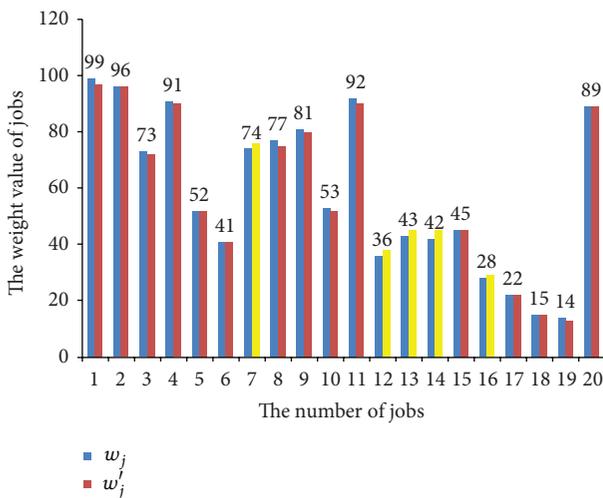


FIGURE 9: The adjustment values of W_j .

IGA can lead to good results and solve the large-sized SMISP problem effectively.

6. Case Studies

6.1. Case Introduction. Shipbuilding manufacturing is an important industry among the largest manufacturing industries in the world. In this paper, we carried out a market research in a shipbuilding workshop which produces sleeves, butt joints (08-6B-CB*56-1983), nuts (01-6B-CB*821-1984), and so on. And it is found that this shop is a typical SMRSP. In this paper, we take 20 jobs as example. All processing parameters of 20 jobs are as follows: p_j ($j = 1, 2, \dots, n$) are processing times and w_j ($j = 1, 2, \dots, n$) are weights. It should be noted that these weights present the costs of producing jobs in this case. Thus, the total weighted

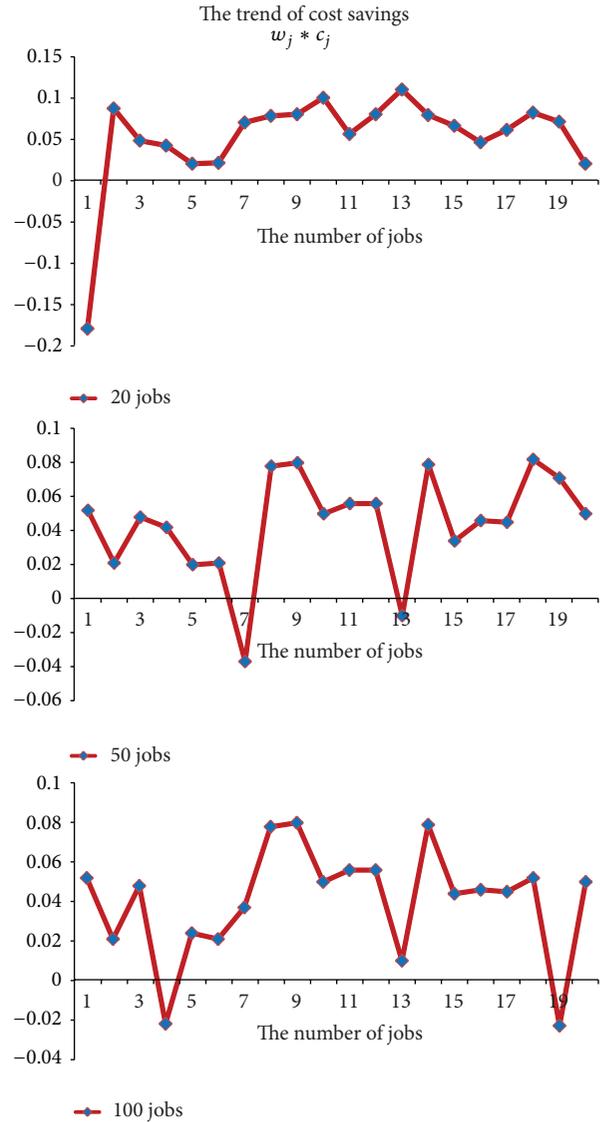


FIGURE 10: The trend of cost savings.

completion time could be considered as production cost or investment. Therefore, minimal adjustment weight is to ensure that the resulting production investment is not greater than the original objective value and thus has very practical significance.

6.2. Case Results. This algorithm was executed under 10 times and a computational result of this case was given: the compute time is 2766 ms and the best solution (the minimal perturbation) is 303. It should be noted that the original weighted completion time is 1260340 (RMB) before inverse scheduling and the corresponding initial job sequence is [1-8, 10-21]. Obviously, from Figure 13, all results here are significantly less than the given target 1260340. Particularly, there exists the optimal solution (the sum of the completion time by IGA) which is 1119230 by testing 10 times. The data indicates that 12% savings are generated for this shipyard. Figure 11 shows

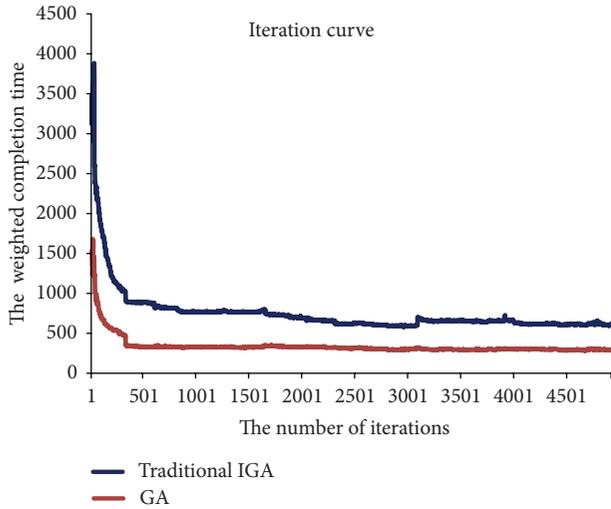


FIGURE 11: The iteration curve of parameter.

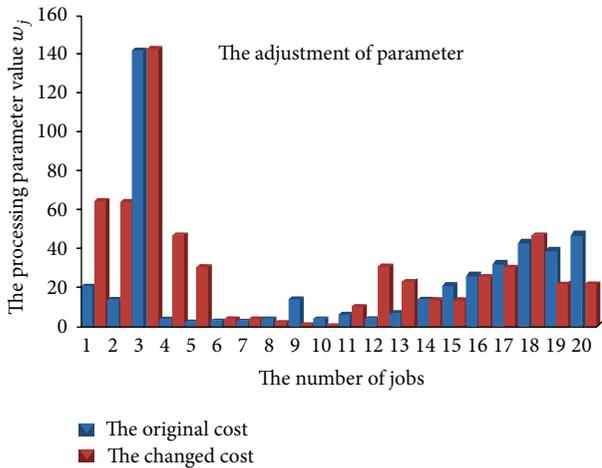


FIGURE 12: The adjustment values of parameter.

convergence curves for both datasets. Figure 12 presents the adjustment values of each weight. Clearly, the method presented in this paper will be useful in this context of SMISP because the method can squeeze out some cost savings. Therefore, this research work has an important guiding sense to a real production.

7. Conclusions and Future Researches

In this paper, this is the first report to propose an improved genetic algorithm (IGA) for the single-machine inverse scheduling problem with the total weight completion time. Firstly, a mathematical model based SMISP is constructed and a novel approach called improved genetic algorithm is employed to solve this problem. The idea of WSPT is taken to develop a local search procedure in IGA, and several adaptations are done to improve the performance of algorithm. We do not only propose a fitness evaluation mechanism to consider various real objectives in SMISP but also investigate



FIGURE 13: Comparison between the results.

the ways such as a feasible initialization method, encoding scheme, and crossover and mutation operators to perfect the algorithm performance. Twenty-seven cases have been used to test the IGA algorithm, and the comparison has been made for this approach and the traditional GA. The experimental results show that the proposed method can generate satisfactory solutions and has achieved significant improvement. Finally, a real shipbuilding production factory is selected as a real-world case study; a saving of 12% is obtainable under the ISMSP.

ISP is complex and lacks a fundamental hypothesis for its research. However, there are still some limitations in research into ISP. First of all, in the most present studies, the current models of ISP formulated above are considered to minimize adjustable processing parameter; however, no research has been done concerning multicriteria problems. With respect to solution approach, linear programming is the most frequently exact procedure used here.

Although ISP is a hard work, its research is necessary. ISP is also very important because it can effectively improve an existing manufacturer system and optimally allocate resources. So, the research on ISP is still a challenge. Many models and methods can be proposed in this area. For these reasons, the researchers will continue to do further and deeper study on ISP. Some suggestions are given here. Most existing models on ISP consider minimally perturbing the job processing time while other common criteria, such as makespan or extra cost, will be studied. In the future, these hybrid intelligence algorithms proposed in previous literatures should be considered to solve ISP for more complex environment.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research work was supported by the National Science Foundation of China (NSFC) under Grant nos. 51375004 and 51121002, Independent Innovation Finance HUST Support Plan under Grant no. 2013TS021, and the National Natural Science Foundation of China Grant no. 61203179.

References

- [1] C. Koulamas, "Inverse scheduling with controllable job parameters," *International Journal of Services and Operations Management*, vol. 1, pp. 35–43, 2005.
- [2] P. Brucker and N. V. Shakhlevich, "Inverse scheduling: two-machine flow-shop problem," *Journal of Scheduling*, vol. 14, no. 3, pp. 239–256, 2011.
- [3] P. Brucker and N. V. Shakhlevich, "Inverse scheduling with maximum lateness objective," *Journal of Scheduling*, vol. 12, no. 5, pp. 475–488, 2009.
- [4] I. Elamvazuthi, P. Vasant, and T. Ganesan, "Hybrid optimization techniques for optimization in a fuzzy environment," *Intelligent Systems Reference Library*, vol. 38, pp. 1025–1046, 2013.
- [5] T. Ganesan, I. Elamvazuthi, K. Z. Ku Shaari, and P. Vasant, "Swarm intelligence and gravitational search algorithm for multi-objective optimization of synthesis gas production," *Applied Energy*, vol. 103, pp. 368–374, 2013.
- [6] A. Kasperski, A. Kurpisz, and P. Zieliński, "Approximating a two-machine flow shop scheduling under discrete scenario uncertainty," *European Journal of Operational Research*, vol. 217, no. 1, pp. 36–43, 2012.
- [7] H. Aissi, C. Bazgan, and D. Vanderpooten, "General approximation schemes for min-max (regret) versions of some (pseudo-)polynomial problems," *Discrete Optimization*, vol. 7, no. 3, pp. 136–148, 2010.
- [8] P. Fattahi and A. Fallahi, "Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability," *CIRP Journal of Manufacturing Science and Technology*, vol. 2, no. 2, pp. 114–123, 2010.
- [9] P. Vasant, "Hybrid LS-SA-PA methods for solving fuzzy non-linear programming problems," *Mathematical and Computer Modelling*, vol. 57, no. 1-2, pp. 180–188, 2013.
- [10] R. K. Ahuja and J. B. Orlin, "Inverse optimization," *Operations Research*, vol. 49, no. 5, pp. 771–783, 2001.
- [11] C. Heuberger, "Inverse combinatorial optimization: a survey on problems, methods, and results," *Journal of Combinatorial Optimization*, vol. 8, no. 3, pp. 329–361, 2004.
- [12] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*, SIAM, Philadelphia, Pennsylvania, 2005.
- [13] L. Z. Wang, "Cutting plane algorithms for the inverse mixed integer linear programming problem," *Operations Research Letters*, vol. 37, no. 2, pp. 114–116, 2009.
- [14] L. Liu and J. Zhang, "Inverse maximum flow problems under the weighted Hamming distance," *Journal of Combinatorial Optimization*, vol. 12, no. 4, pp. 395–408, 2006.
- [15] L. L. Liu, C. T. Ng, and T. C. E. Cheng, "Bicriterion scheduling with equal processing times on a batch processing machine," *Computers & Operations Research*, vol. 36, no. 1, pp. 110–118, 2009.
- [16] R. J. Chen, F. Chen, and T. G. Chun, "Inverse problems of a single machine scheduling to minimize the total completion time," *Journal of Shanghai Second Polytechnic University*, vol. 22, no. 2, pp. 1–7, 2005.
- [17] R. J. Chen and G. C. Tang, "Inverse problems of supply chain scheduling and flowshop scheduling," *Operations Research and Management Science*, vol. 18, no. 2, pp. 80–84, 2009.
- [18] H. Pham and X. Lu, "Inverse problem of total weighted completion time objective with unit processing time on identical parallel machines," *Journal of East China University of Science and Technology*, vol. 38, no. 6, pp. 757–761, 2012.
- [19] S. Batun and M. Azizolu, "Single machine scheduling with preventive maintenances," *International Journal of Production Research*, vol. 47, no. 7, pp. 1753–1771, 2009.
- [20] B. S. Kim and C. M. Joo, "Single-machine total completion time scheduling with position-based deterioration and multiple rate-modifying activities," *Industrial Engineering Management Systems*, vol. 10, no. 4, pp. 247–254, 2011.
- [21] P. Liu and X. Tian, "Two-agent single-machine scheduling with resource-dependent starting times," *Mathematical Problems in Engineering*, vol. 2013, Article ID 805261, 5 pages, 2013.
- [22] D.-C. Li, P.-H. Hsu, and C.-C. Chang, "A genetic algorithm-based approach for single-machine scheduling with learning effect and release time," *Mathematical Problems in Engineering*, vol. 2014, Article ID 249493, 12 pages, 2014.
- [23] S. Balin, "Non-identical parallel machine scheduling using genetic algorithm," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6814–6821, 2011.
- [24] F. T. S. Chan, K. L. Choy, and Bibhushan, "A genetic algorithm-based scheduler for multiproduct parallel machine sheet metal job shop," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8703–8715, 2011.
- [25] C.-J. Hsu, "Single-machine scheduling with aging effects and optional maintenance activity considerations," *Mathematical Problems in Engineering*, vol. 2013, Article ID 634503, 6 pages, 2013.
- [26] I. Karaoglan and F. Altiparmak, "A hybrid genetic algorithm for the location routing problem with simultaneous pickup and delivery," *Industrial Engineering & Management Systems*, vol. 10, no. 1, pp. 24–33, 2011.
- [27] K. Rahmani and I. Mahdavi, "A genetic algorithm for the single machine preemptive scheduling problem with linear earliness and quadratic tardiness penalties," *International Journal of Advanced Manufacturing Technology*, vol. 65, no. 5–8, pp. 763–770, 2013.
- [28] X. Yu, Y. Zhang, and K. Huang, "Minimizing the makespan for scheduling problems with general deterioration effects," *Mathematical Problems in Engineering*, vol. 2013, Article ID 218981, 8 pages, 2013.
- [29] C. M. Joo and B. S. Kim, "Genetic algorithms for single machine scheduling with time-dependent deterioration and rate-modifying activities," *Expert Systems with Applications*, vol. 40, no. 8, pp. 3036–3043, 2013.
- [30] S. H. Chen, C. M. Chen, and Y. C. Liou, "Artificial chromosomes with genetic algorithm 2 (ACGA2) for single machine scheduling problems with sequence-dependent setup times," *Applied Soft Computing*, vol. 17, pp. 167–175, 2014.
- [31] P. Vasant, "Hybrid Mesh adaptive direct search genetic algorithms and line search approaches for fuzzy optimization problems in production planning," in *Handbook of Optimization*, vol. 38 of *Intelligent Systems Reference Library*, pp. 779–799, Springer, Berlin, Germany, 2013.

- [32] P. C. Chang, J. C. Hsieh, and S. G. Lin, "The development of gradual-priority weighting approach for the multi-objective flowshop scheduling problem," *International Journal of Production Economics*, vol. 79, no. 3, pp. 171–183, 2002.
- [33] P. C. Chang, J. C. Hsieh, and Y. W. Wang, "Genetic algorithms applied in BOPP film scheduling problems: minimizing total absolute deviation and setup times," *Applied Soft Computing Journal*, vol. 3, no. 2, pp. 139–148, 2003.
- [34] R. Cheng, M. Gen, and Y. Tsujimura, "A tutorial survey of job-shop scheduling problems using genetic algorithms - I. Representation," *Computers and Industrial Engineering*, vol. 30, no. 4, pp. 983–997, 1996.
- [35] R. Cheng, M. Gen, and Y. Tsujimura, "Tutorial survey of job-shop scheduling problems using genetic algorithms: part II. Hybrid genetic search strategies," *Computers and Industrial Engineering*, vol. 37, no. 1, pp. 51–55, 1999.
- [36] S. Koh, P. Koo, D. Kim, and W. Hur, "Scheduling a single batch processing machine with arbitrary job sizes and incompatible job families," *International Journal of Production Economics*, vol. 98, no. 1, pp. 81–96, 2005.
- [37] W. E. Smith, "Various optimizers for single-stage production," *Naval Research Logistics Quarterly*, vol. 3, pp. 59–66, 1956.
- [38] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.
- [39] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.
- [40] C. R. Reeves, "Genetic algorithms for the operations researcher," *INFORMS Journal on Computing*, vol. 9, no. 3, pp. 231–250, 1997.
- [41] C. Reeves, "Genetic algorithms," in *Handbook of Metaheuristics*, F. Glover and G. A. Kochenberger, Eds., pp. 55–82, Kluwer Academic, Dordrecht, The Netherlands, 2003.
- [42] O. Ghrayeb and P. Damodaran, "A hybrid random-key genetic algorithm to minimize weighted number of late deliveries for a single machine," *International Journal of Advanced Manufacturing Technology*, vol. 66, no. 1–4, pp. 15–25, 2013.
- [43] T. Chiang, H. Cheng, and L. Fu, "NNMA: an effective memetic algorithm for solving multiobjective permutation flow shop scheduling problems," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5986–5999, 2011.
- [44] D. Jakobovic and L. Budin, "Dynamic scheduling with genetic programming," in *Genetic Programming*, vol. 3905 of *Lecture Notes in Computer Science*, pp. 73–84, 2006.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

