Design of a genetic algorithm for bi-objective flow shop scheduling problems with re-entrant jobs

C.K.M.Lee[1], Danping Lin[1], William Ho[2], Zhang Wu[1]

C.K.M.Lee[1], Danping Lin[1], William Ho[2], Zhang Wu[1]

[1]Division of Systems and Engineering Management, School of Mechanical and Aerospace Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798

Tel: +65 67906891 +65 83590052 +65 67904445

Email: CKMLee@ntu.edu.sg,LIND0020@ntu.edu.sg,mzwu@ntu.edu.sg

[2]Operations & Information Management Group, Aston Business School, Aston University, Aston Triangle, Birmingham B4 &ET,U.K.

Tel: +44 (0) 121 204 3000

Email: w.ho@aston.ac.uk

Abstract

This paper presents a simulated genetic algorithm (GA) model of scheduling the flow shop problems with re-entrant jobs. The objectives of this research are to minimize the weighted tardiness and makespan. The proposed model considers that the jobs with non-identical due dates are processed on the machines in the same order. Furthermore, the re-entrant jobs are stochastic as only some jobs are required to reenter to the flow shop. The tardiness weight is adjusted once the jobs re-enter to the shop. The performance of the proposed GA model is verified by a number of numerical experiments where the data come from the case company. The results show the proposed method has a higher order satisfaction rate than the current industrial practices.

Keyword: bi-objective, re-entrant, genetic algorithm, flow shop

1. Introduction

The Flow Shop Scheduling Problem (FSP) has an extensive background in industrial applications and has attracted many researchers' attention since it was proposed by Johnson (1954). Most research papers assume that the jobs will not re-enter the processing line once they are finished. While this assumption simplifies the analysis or represents only certain industrial applications, it does not consider the condition that perhaps some fault jobs need to be reworked after quality control in real environment.

In a manufacturing environment, production problems that are related to the re-entrant jobs can be divided into two important categories: (1) quality problem (2) specification requirement. For example, at a repair facility where coating is needed, a rework job is incurred for blasting the part again whenever a new crack is found. Regardless of the job category, once the jobs are reworked, delivery schedule is tighter.

To tackle the complexity and uncertainty in this situation, analytic models are proposed but the model may be over-simplified, otherwise heavy computational effort is required. Therefore, simulation can overcome this shortcoming and be used in FSP solution. As a test and validation tool, simulation model can only evaluate a given design instead of providing more general decision making function. However, combining the simulation thread and the genetic algorithm optimization instead of using rudimentary optimization techniques, not only the intelligent decision making of the simulation is enhanced, but also the complex system model would have broader applicability (Paul and Chanev, 1998). Therefore, simulation method for flow shop scheduling problems is proposed in this paper.

In this paper, a genetic algorithm is suggested to solve a flow shop with stochastic jobs reenter for the bi-objective of minimizing the weighted tardiness and makespan. As this kind of problems deprived (?) from the real manufacturing line with special due-date requirement, the experiment data are adopted from real case as well. The paper is organized as follows. Section 2 presents a brief overview of the extend model and related literature. Section 3 introduces the problem formulation. The proposed hybrid GA is developed in Section 4. The performance of the method is reported in Section 5, and finally Section 6 covers the conclusion.

2. Literature review

After the publication of Johnson's classical paper on the flow shop scheduling problems, few optimization techniques are available in the early decades. The techniques are mainly mathematical programming

(S.Manne, 1960, Wagner, 1959), combinatorial approaches (N.D.Gupta, 1969, N.D.Gupta, 1979, Z.A.Lomnicki, 1965) because of the lack of computer power and efficient computer programs. And most of the objective functions are limited to minimize the makespan. Only after the emergence of the theory of NP-hardness (Garey and S.Johnson, 1979) and the high-speed computational programming, the research of flow shop problem began to develop faster. It does not only broaden scope such as multi-machine consideration, multi-criteria in objective function, or relaxation in the assumption of setup time and etc., but also adopts new techniques like meta-heuristics and artificial intelligence based techniques. Among all these techniques, Genetic Algorithm (GA) has promising performance for optimization (Zhang et al., 2006, Zhu et al., 2008, Yang et al., 2008, Lin-Yu and Ya-Tai, 2009). The GA can be used alone or combined with other methods like local search or features that are embedded within other methods like the work of Nearchou (2004). However, no matter how the GA is used, the majority of the papers that used GA for flow shop problems only consider a single objective like makespan (Caraffa et al., 2001, Iyer and Saxena, 2004, Marimuthu and Ponnambalam, 2005, Nagano et al., 2008, Rajkumar and Shahabudeen, 2009, Shih Hsin et al., 2009) or tardiness (Iima and Sannomiya, 1995, Al-Anzi and Al-Fares, 2001, Swaminathan et al., 2004). Even though some researchers considered the bi-criteria or multiple criteria, they simply converted several objectives into a single one by using the weighted allocation. Framinan (2009) proposed two new weighting schemes to aggregate two objectives (makespan and tardiness) into a scalar function for the flow shop scheduling. In an extensive ANOVA test, local search is embedded for all offsprings that belong to the current population and all the data are analyzed under various parameter setting. The results indicated that the proposed method outperforms other weighted mechanism to produce more homogeneous set of solution to the problem. Mahadvi et al.(2008) also used weighted sum to tackle the *s*-stage flow shop with serial batch production at the last stage, where multiple objectives are addressed for minimizing the total weighted earliness, the total weighted tardiness and the total weighted waiting time. Other related finding can be found from (Ruiz and Allahverdi, 2009, Ishibashi et al., 2000, Yandra and Tamura, 2007).

Traditionally , the bi-criteria or multiple criteria are treated one by one, namely , problems are first solved for the first criterion by ignoring others, and then solved the second criterion under the first constraint space. This method is developed by Neppalli et al.(1996) whoproposed two GA-based approaches to solve a two-stage flow shop with the objective of minimizing the total flow time subject so as to obtain the optimal makespan. In our paper, we will use this method as reference to take both makespan and tardiness into consideration.

Relaxing or adding some of the assumed constraints can bring the FSP research direction into a completely new domain, such as no-wait flowshop, batch/lot flowshop, or bottleneck flowshop. One particular type of flow shop is the re-entrant flowshop, wherein some jobs will enter the manufacturing line more than once, and this situation is common in real industrial environment. To solve this problem, two methods are proposed so far: one is to transfer the re-entrant problem into the non-re-entrant one (Jing et al. 2008). As the sub-jobs of the original job do not need to be processed successively, when every sub-job (?) has the same time of entering the manufacturing line (say, *L* level), re-entrant problems with *n* sub-jobs(?) is changed into non-re-entrant problem with *nL* sub-jobs. The other method is adding one no-passing constraint which means no job is allowed to pass any former job. Improved heuristic methods can also be developed to deal with the re-entrant problem (Choi and Kim, 2008, Choi and Kim, 2009, Chu et al., 2009). Pan and Chen (2003) proposed three extended mixed Binary Integer Programming formulations and six extended effective heuristics to tackle this problem with the aim of minimizing the makespan. Chen et al.(2008) applied hybrid tabu search to solve the problem in order to minimize makespan. However, both of these methods assumed that all the jobs visit certain machine more than once. In addition, they only pay attention to one objective, mostly aiming at minimizing the makespan. However, in the real situation, the production engineers need to consider more than one objective. Therefore, bi-objective model is proposed in this paper to represent the real situation where the two objectives are interrelated. As the probability of the re-entrant is not fixed for each job and the no-passing constraint is loosen, the proposed method is the extension of the model of Pan and Chen(2003).

3. Problem formulation

Suppose there are n jobs, J_1, J_2,… J_n and m machines, M_1, M_2,… M_m in the shop. Job i has N_i operations and the processing time of its j^{th} operation P_{ij} is deterministic and prescribed in advance. For job i, its P_{ij+1} must be initiated right after the completion of P_{ij}. Normally in many cases customers want to receive their orders on time. In a case of any delay, there is no benefit for customers that may result in unsatisfaction and loss of customers. Indeed, the tardiness is an important attribute of service quality, and a customer's dissatisfaction tends to increase quadratically with the tardiness, as proposed in the loss function of Taguchi (2005). From the industrial point of view, tardiness means rush shipping costs, lost sales and loss goodwill. Thus it is necessary to deliver the orders on the right day at the right volume. The minimization of the total tardiness is considered as the main objective in the scheduling problems. In addition, minimizing the makespan is commonly studied in scheduling problems by researchers as it leads to the reduction in the total Work-In-Process (WIP) inventories, and the minimization of irregularities and inordinate shop flow crowding due to uncompleted jobs. This paper proposes a novel bi-objective model that minimizes the tardiness and makespan at the same time.

a. Problem assumptions

Except the normal assumptions that concern the jobs, machines and processing policy as shown in the work of Gupta and Stafford Jr (2006), specific assumptions would be chosen throughout this paper:

Even though each operation is independent of each other in the same job, we assume that the operations of the same job are processed successively. The current jobs cannot be interrupted by operations of other jobs. This condition, which is more related to the real world, can greatly reduce the WIP and the operators do not need to frequently shift work. The jobs follow the same sequence but they do not pass all the machines.

b. Proposed programming model

The variables are introduced and defined as following.

Problem parameters:

N: the set of all jobs

n: the number of jobs

i, j: job index (j is operation index on page 5?)

m: machine index (m is the number of machines on page 5?)

$\mathrm{M_i}$: the set of machines to process job i

$\mathrm{E_m}$: the set of jobs that might be processed on machine m

B: the set of pairs of jobs which have precedence relationship

$\mathrm{p_i}$: processing time of job i

Decision variables:

$\mathrm{x_{im}} = 1$, if operation of job i is assigned to machine m; 0, otherwise;

$\mathrm{y_{ijm}} = 1$, if operation of job i and j are assigned to the same machine m; 0, otherwise;

$\mathrm{z_{ijm}} = 1$, if operation of job i immediately precedes j on machine m;

$\mathrm{t_i}$ is the starting time of job i ;

$\mathrm{C_{max}}$ is the completion time of the last job.

i. Model for phase 1

In the first phase, model is set to minimize the total tardiness which is calculated as the sum of the product of the tardiness weight W_i and the tardy time T_i. The tardiness weight is related to the tardy time. If tardy time is within half a month, then the tardiness weight is considered as tight, given a value of 4. If tardy time is within 30 days or within 45 days, the tardiness weight is recognized as moderate or loose, given the value of 2 or 1 respectively.

Following is a model for Phase 1

$$\mathrm{Min} \sum_{i=1}^{N} W_i T_i \tag{1}$$

$$\text{s.t.} \sum_{\mathrm{m \in M_i}} \mathrm{x_{im}} = 1 , \forall \mathrm{i} \in \mathrm{N} \tag{2}$$

$$\mathrm{t_i} \geq 0, \forall \mathrm{i} \in \mathrm{N} \tag{3}$$

$$\mathrm{t_i} + \mathrm{p_i} \leq \mathrm{t_{i+1,}}\ \forall \mathrm{i} \in \mathrm{N} \tag{4}$$

$$\mathrm{y_{ijm}} = \mathrm{y_{jim}}, \forall \mathrm{i,j} \in \mathrm{E_m}, \forall \mathrm{m} \in \mathrm{M_i} \tag{5}$$

$$\mathrm{y_{ijm}} \leq 0.5\left(\mathrm{x_{im}} + \mathrm{x_{jm}}\right) \leq \mathrm{y_{ijm}} + 0.5, \quad \forall \mathrm{i,j} \in \mathrm{E_m}, \forall \mathrm{m} \in \mathrm{M_i} \tag{6}$$

$$\sum_{\mathrm{j \in E_m}} \mathrm{Z_{ijm}} \leq 1, \forall \mathrm{i,j} \in \mathrm{E_m}, \forall \mathrm{m} \in \mathrm{M_i} \tag{7}$$

$$\sum_{\mathrm{j \in E_m}} \mathrm{Z_{jim}} \leq 1, \forall \mathrm{i,j} \in \mathrm{E_m}, \forall \mathrm{m} \in \mathrm{M_i} \tag{8}$$

$$\mathrm{z_{ijm}} + \mathrm{z_{jim}} \leq 1, \forall \mathrm{i,j} \in \mathrm{E_m}, \forall \mathrm{m} \in \mathrm{M_i} \tag{9}$$

$$\mathrm{t_i} \leq \mathrm{t_j}, \forall \mathrm{i,j} \in \mathrm{B} \tag{10}$$

$$\mathrm{x_{im}}, \mathrm{y_{ijm}}, \mathrm{z_{ijm}} = 0 \text{ or } 1, \forall \mathrm{i,j} \in \mathrm{N}, \forall \mathrm{m} \in \mathrm{M_i} \tag{11}$$

Where $T_i = \max(0, F_i - D_i)$

$W_i = [1,2,4]$

F_i=Finish date of final job (date)

D_i=Due date of final job (date)

Eq. (1) minimizes the main objective, the total tardiness of jobs. Constraints (2) ensure that each operation must be processed by exactly one machine. Constraints (3) ensure that each operation begins after time zero. Constraints (4) ensure that the order of operation of each job is respected. Constraints (5) and (6) ensure that $y_{ijm} = y_{jim} = 1$ when $x_{ijm} = x_{jim} = 1$. Constraints (7) and (8) ensure that each operation has at most one predecessor and successor on machine m. Constraint (9) ensures that z_{ijm} and z_{jim} cannot equal to 1 simultaneously. Constraint (10) determines the set of pairs of jobs between which there is a precedence relationship. Constraint (11) ensures x_{im}, y_{ijm}, z_{ijm} are binary constraints.

ii. Model for phase 2

After solving the model for the first phase, the optimal solution obtained by minimizing the total tardiness is considered as an additional constraint to the second phase that is to minimizing makespan. It is also assumed that T is a solution that minimizes the tardiness. Following is a model for Phase 2.

$$\text{Min} C_{max} = \text{max}_i(t_i + p_i) \quad (12)$$

s.t. Constraints 2-11,

$$\sum_{i=1}^{N} T_i = T \quad (13)$$

Eq. (12) minimizes the total makespan of jobs. Constraint (13) guarantees that the tardiness obtained at the first phase remains at its optimum amount. The solution obtained in the second phase is the final solution for the minimization of jobs tardiness and total makespan.

4. A hybrid genetic algorithm

Genetic Algorithm (GA) is an adaptive method that can be used to solve optimization problems. GA is based on the genetic process of biological organism, in which the fitness of individual determines its ability to survive and reproduce. The term was first used by Holland (1992) in his book "Adaptation in natural and artificial systems". GA mechanism starts by encoding the problem to produce a list of genes, which are then randomly combined to produce a population of chromosomes. Each chromosome represents a possible solution. Genetic operations work on the chromosomes to produce offspring. The fitness of these chromosomes is then measured and the probability of their survival is determined by their fitness (Pongcharoen et al., 2002).

The pseudo-code is listed as below:

BEGIN

```
  Generate initial population;
  Compute fitness of each individual;
  REPEAT /* New generation /*
    FOR population_size / 2
      Select two parents from old generation;
       /* biased to the  ones with higher fitness */
      Recombine parents for two offspring;
      Compute fitness of offspring;
      Insert offspring in new generation
    END FOR
  UNTIL population has converged
END
```

a. Chromosome representation

The job sequences are used as chromosomes. For example, if there are 4 jobs waiting in line, they are denoted as 1,2,3,4. Then the chromosome is expressed as [4, 1, 3, 2], which means that job 4 is processed first and then followed by job 1 and so on.

b. Initial population

Usually the population is randomly generated at a constant amount. Obviously, a constant population size is not suitable for problems with different sizes. The convergence is too slow for a large population while the optimal solution is very difficult to obtain for a small population. Then it is assumed that the population size is a function relates to the number n of jobs, Pop_size=inte $[1000 * (1 - e^{-0.001n})] + 40$,where inte[.] is an operator that rounds a real number to the nearest integer. For example, Pop_size is 60 if n is 20, while Pop_size is 80 if n is 40.

c. Selection

According to Holland (1992), selection is made to obtain the parents for the processing of all genetic operator. A chromosome with a higher fitness value will have a higher probability of being reproduced. A heuristic method is adopted to improve the iteration where one parent is chosen as per a parameter, selection_rate (the probability of a chromosome being selected is in proportion to its ranking within the population), and another parent is randomly generated. If the offspring are better than their parents, the worst parents would be replaced.

d. Crossover and mutation

The purpose of crossover is to exchange information between randomly selected parent chromosomes to produce offspring for the next generation which retains the good properties of the parent (Chang et al., 2007). Usually there are three kinds of crossover operators, named one-point, two-point, and uniform crossover. The one-point and two-point crossover operators are the same at randomly selected points of the two parent parameter sets but differ by the number of cuts made on the parent chromosomes as shown in Figure 1, while the uniform crossover operates on randomly selected individual genes instead of parts of the chromosome. In this research, the two-point crossover operator with a fixed crossover rate of P_c(say 0.8) is employed for reproducing the offspring during each iteration.

(Insert Figure 1 about here)

The mutation operator is used to prevent the crossover from getting stuck in a local optimum. During the mutation process, the mutation rate increases as the diversity of the population decreases in order to broaden the search space. The mutation rate is calculated by multiplying the initial_mutation_rate (set less than 5%, say 3%) with a factor F (ranging from 1 to 5) if the measured diversity is less than an acceptable level (say 10%). The measure of diversity is defined as $\frac{f_{max}-f_{min}}{\bar{f}}$, where f_{max}, f_{min}, $\bar{f}$ are the largest, smallest and average fitness values respectively.

e. Stopping criteria

According to previous experiment, the model will converge to the optimal solution within 200 iterations, thus the offspring generation is set to terminate when the maximum number of iteration reaches 200.

f. Simulator

In the hybrid algorithm, the heuristic is implemented via a simulation operator. As each job is denoted with an index, the operator is designed to check whether a j0b is a re-entered one by its index before each computational permutation. If it is imperative, the weight of that job would be increased because of the tight due date level. The overall process is showed in the following Figures 2 and 3.

(Insert Figure 2 and 3 about here)

5. Performance of the hybrid genetic algorithm

In this section, the effectiveness of the hybrid algorithm is evaluated using real case data.

a. An illustrative example

In order to evaluate the performances of the proposed heuristics, based on the example of an energy service company with typical flow shop, experiments on randomly generated problems with 20, 30, and 40 jobs are conducted. (Even though the experiment with larger size problems can be dealt with, there is no real case data available for comparison) This case company repairs the heavy machines based on the customer's order. One repair job has several operations, but the main process order is the same, only with some exceptions in the difference in the processing time in a particular stage and in that some jobs do not pass all the machines. Thus the processing times are taken as randomly distributed in the ranges [0, 15].Even though each job has an exact due date, a period is given around the exact due date. It is necessary to judiciously design the due date distribution as the tightness level of the due date is important to determine the hardness of the problem (Baker, 1974). According to Pinedo and Singer(1999), 20% of customers are very important; 60% are average and the remaining 20% are of less important. Hence, for weighted tardiness problems, three levels of due-date tightness are classified: tight (20%), moderate (60%) and loose (20%) with a weight of 4, 2 and 1 respectively (Zhou et al., 2009). Therefore the due dates are also uniformly distributed on three different ranges as [5, 15], [16, 30] and [31, 45]. As GA is a stochastic searching heuristics, the result of every test instance is unlikely to be the same. In order to compare the average performance, for each job size, 10 runs are performed. The experiment results are shown in Table 1. According to the results of the proposed method, the mean of the total makespan of each problem size is decreased as job increases but the proposed model always performs better than the industrial practice. The proposed method outperforms not only in terms of average deviation of makespan, but also in the order of satisfaction rate which means the percentage of jobs being fulfilled on time. But with the increase of the problem size, the decrease rate of order satisfaction in practical situation is steeper compared with the proposed model. With regard to the practical maximum deviation of makespan, it almost increases doubly. It is difficult to calculate manually with the large number of jobs. However, it is noticed that even though the proposed method is better than the current approach , 5% (23%-18%) (it is difficult to understand "5% (23%-18%)") of order satisfaction can be reached. If 5% is multiplied by the revenue of each project, the amount shall be prominent

(Insert Table 1 about here)

Table 2 considers the condition of re-entrant jobs. Once detecting the re-entrant jobs, the system would adjust them to a higher weight and join the scheduling line. To simplify the flow shop scheduling problem, the original weight of the re-entrant is always set as “1”. In the real world, rework is usually not taken into consideration. However, once the re-entrant jobs come, the schedulers will postpone all the current jobs and deal with the re-entrant jobs. This way may help to satisfy the due date tightness of the re-entrant jobs, but it would lead to delay for the subsequent jobs. For the data of 20 jobs, there is not much performance difference between the practical approach and the proposed method. However, if only 10 more jobs are added, the mean deviation of makespan of the practical method increased sharply. When more re-entrant jobs come in, the fluctuation is more serious. It can be realized from the column of the practical method in Table 2. To the opposite, in the proposed method, even with the increase of re-entrant jobs, the order satisfaction rate is kept at an acceptable level above 10%.

(Insert Table 2 about here)

According to the historical records, the re-entrant jobs are about 5% of total task. Take the problem size of 20 jobs as example. If re-entrant does not happens, the total order satisfaction rate is 23%. If one re-entrant (3%) does come, the order satisfaction rate for 19 jobs is 21%, and two re-entrant (2%) come with order satisfaction rate of 21%. Then the total satisfaction rate is 3%*21%+2%*21%+(1-3%-2%)*23%=22.90%. Similarly the distribution is showed in Table 3.

(Insert Table 3 about here)

6. Conclusions

In this paper, a genetic algorithm was applied to flow shop scheduling with re-entrant jobs to minimize the tardiness and makespan. For minimizing the tardiness, the key is to set different due date tightness levels, especially for the re-entrant jobs; the tightness level is adjusted to the highest automatically. The scheduling permutation is dependent on the tardiness, while the heuristics approach depends on due date. Real world case was studied conducted to examine the effectiveness of the algorithm with respect to different levels of due date tightness. The results obtained by the proposed method are better than the practical method. This study reveals that even a small advancement can bring in significant revenue in the real life. The future work may lie in the parameter setting such as crossover method, mutation rate, stopping criteria to improve the performance of the genetic algorithm to achieve higher order satisfaction rate. The study of problems with multiple objectives is also an important direction. In addition, because of the extensive application in the manufacturing environment, more company cases with less assumptions and constraints may be considered. The significance of this study is to pay the way for multi-objective functions in production scheduling, thereby allow production engineers to realize the tradeoff for re-entrant flow shop scheduling.

Acknowledgements

The authors would like to acknowledge the support of Mr. Teo from GE-Kepple Singapore and the Academic Research Fund (AcRF) Tier1 Grant from Nanyang Technological University Research Committee.

References

AL-ANZI, F. S. & AL-FARES, M. S. 2001. Minimizing lateness of multimedia data object requests using genetic algorithms. *Advances in Automation, Multimedia and Video Systems, and Modern Computer Science*, 174-179.

BAKER, K. R. 1974. *Introduction to sequencing and scheduling,* New York, NY, Wiley.

CARAFFA, V., IANES, S., BAGCHI, T. P. & SRISKANDARAJAH, C. 2001. Minimizing makespan in a blocking flowshop using genetic algorithms. *International Journal of Production Economics,* 70**,** 101-15.

CHANG, J.-L., GONG, D.-W. & MA, X.-P. 2007. A Heuristic Genetic Algorithm for No-Wait Flowshop Scheduling Problem. *Journal of China University of Mining and Technology,* 17**,** 582-586.

CHEN, J.-S., PAN, J. C.-H. & WU, C.-K. 2008. Hybrid tabu search for re-entrant permutation flow-shop scheduling problem. *Expert Systems with Applications,* 34**,** 1924-1930.

CHOI, S.-W. & KIM, Y.-D. 2008. Minimizing makespan on an m-machine re-entrant flowshop. *Computers & Operations Research,* 35**,** 1684-1696.

CHOI, S.-W. & KIM, Y.-D. 2009. Minimizing total tardiness on a two-machine re-entrant flowshop. *European Journal of Operational Research,* 199**,** 375-384.

CHU, F., CHU, C. & DESPREZ, C. 2009. Series production in a basic re-entrant shop to minimize makespan or total flow time. *Computers & Industrial Engineering,* In Press, Corrected Proof.

FRAMINAN, J. M. 2009. A fitness-based weighting mechanism for multicriteria flowshop scheduling using genetic algorithms. *International Journal of Advanced Manufacturing Technology,* 43**,** 939-948.

GAREY, M. & S.JOHNSON, D. 1979. *Computers and intractability : a guide to the theory of NP-completeness*, W.H. Freeman and Company.

GUPTA, J. N. D. & STAFFORD JR, E. F. 2006. Flowshop scheduling research after five decades. *European Journal of Operational Research,* 169**,** 699-711.

H.HOLLAND, J. 1992. *Adaptation in natural and artificial systems*, MIT Press Cambridge,MA,USA.

IIMA, H. & SANNOMIYA, N. 1995. The influence of lethal gene on the behavior of genetic algorithm. *Transactions of the Society of Instrument and Control Engineers,* 31**,** 569-76.

ISHIBASHI, H., AGUIRRE, H. E., TANAKA, K. & SUGIMURA, T. Year. Multi-objective optimization with improved genetic algorithm. *In:* Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, 8-11 Oct. 2000, 2000 Piscataway, NJ, USA. IEEE, 3852-7.

IYER, S. K. & SAXENA, B. 2004. Improved genetic algorithm for the permutation flowshop scheduling problem. *Computers and Operations Research,* 31**,** 593-606.

JING, C., TANG, G. & QIAN, X. 2008. Heuristic algorithms for two machine re-entrant flow shop. *Theoretical Computer Science,* 400**,** 137-143.

JOHNSON, S. M. 1954. Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics,* Quarterly 1**,** 61-68.

LIN-YU, T. & YA-TAI, L. 2009. A hybrid genetic local search algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research,* 198**,** 84-92.

MAHDAVI, I., MOJARAD, M. S., JAVADI, B. & TAJDIN, A. Year. A genetic approach for solving a hybrid flow shop scheduling problem. *In:* 2008 IEEE International Conference on Industrial Engineering and Engineering Management, IEEM 2008, December 8, 2008 - December 11, 2008, 2008 Singapore, Singapore. Inst. of Elec. and Elec. Eng. Computer Society, 1214-1218.

MARIMUTHU, S. & PONNAMBALAM, S. G. 2005. Heuristic search algorithms for lot streaming in a two-machine flowshop. *International Journal of Advanced Manufacturing Technology,* 27, 174-180.

N.D.GUPTA, J. 1969. A general algorithm for the n x m flowshop scheduling problem. *The International Journal of Production Research*, 241-247.

N.D.GUPTA, J. 1979. M-stage flowshop by branch and bound. *Opsearch,* 9, 37-43.

NAGANO, M. S., RUIZ, R. & LORENA, L. A. N. 2008. A constructive genetic algorithm for permutation flowshop scheduling. *Computers & Industrial Engineering,* 55, 195-207.

NEARCHOU, A. C. 2004. Flow-shop sequencing using hybrid simulated annealing. *Journal of Intelligent Manufacturing,* 15, 317-28.

NEPPALLI, V. R., CHEN, C.-L. & GUPTA, J. N. D. 1996. Genetic algorithms for the two-stage bicriteria flowshop problem. *European Journal of Operational Research,* 95, 356-373.

PAN, J. C.-H. & CHEN, J.-S. 2003. Minimizing makespan in re-entrant permutation flow-shops. *Journal of the Operational Research Society,* 54, 642-653.

PAUL, R. J. & CHANEV, T. S. 1998. Simulation optimisation using a genetic algorithm. *Simulation Practice and Theory,* 6, 601-611.

PINEDO, M. & SINGER, M. 1999. A shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop. *Naval Research Logistics,* 46, 1-17.

PONGCHAROEN, P., HICKS, C., BRAIDEN, P. M. & STEWARDSON, D. J. 2002. Determining optimum Genetic Algorithm parameters for scheduling the manufacturing and assembly of complex products. *International Journal of Production Economics,* 78, 311-322.

RAJKUMAR, R. & SHAHABUDEEN, P. 2009. An improved genetic algorithm for the flowshop scheduling problem. *International Journal of Production Research,* 47, 233-249.

RUIZ, R. & ALLAHVERDI, A. 2009. Minimizing the bicriteria of makespan and maximum tardiness with an upper bound on maximum tardiness. *Computers & Operations Research,* 36, 1268-1283.

S.MANNE, A. 1960. On the Job Shop Scheduling Problem. *Operations Research,* 8, 219-223.

SHIH HSIN, C., PEI CHANN, C. & QINGFU, Z. Year. A self-guided genetic algorithm for flowshop scheduling problems. *In:* 2009 IEEE Congress on Evolutionary Computation (CEC 2009), 18-21 May 2009, 2009 Piscataway, NJ, USA. IEEE, 471-8.

SWAMINATHAN, R., FOWLER, J. W., PFUND, M. E. & MASON, S. J. Year. Minimizing total weighted tardiness in a dynamic flowshop with variable processing times. *In:* IIE Annual Conference and Exhibition 2004, May 15, 2004 - May 19, 2004, 2004 Houston, TX, United states. Institute of Industrial Engineers, 819.

TAGUCHI, G. 2005. *Taguchi's quality engineering handbook*, John Wiley.

WAGNER, H. M. 1959. An integer linear-programming model for machine scheduling. *Naval Research Logistics Quarterly,* 6, 131-140.

YANDRA & TAMURA, H. 2007. A new multiobjective genetic algorithm with heterogeneous population for solving flowshop scheduling problems. *International Journal of Computer Integrated Manufacturing,* 20, 465-77.

YANG, N., LI, X.-P., ZHU, J. & WANG, Q. Year. Hybrid Genetic-VNS algorithm with total flowtime minimization for the no-wait flowshop problem. *In:* 7th International Conference on Machine Learning and Cybernetics, ICMLC, July 12, 2008 - July 15, 2008, 2008 Kunming, China. Inst. of Elec. and Elec. Eng. Computer Society, 935-940.

Z.A.LOMNICKI 1965. A branch and bound algorithm for the
exact solution of the three machine scheduling problem. *Operational Research Quarterly,* 16, 89-100.

ZHANG, L., WANG, L. & ZHENG, D.-Z. 2006. An adaptive genetic algorithm with multiple operators for flowshop scheduling. *International Journal of Advanced Manufacturing Technology,* 27**,** 580-587.

ZHOU, H., CHEUNG, W. & LEUNG, L. C. 2009. Minimizing weighted tardiness of job-shop scheduling using a hybrid genetic algorithm. *European Journal of Operational Research,* 194**,** 637-649.

ZHU, X., LI, X. & WANG, Q. 2008. Objective increment based metaheuristic for total flowtime minimization in no-wait flowshops. *Journal of Southeast University (English Edition),* 24**,** 168-173.

Table 1 Comparison of the heuristics with the practical solution

Problem size	Performance measure	Practical	Proposed
20 jobs	Average deviation	59.68	28.13
	Maximum deviation	61.9	40.45
	Order satisfaction rate	18%	23%
30 jobs	Average deviation	72.24	61.32
	Maximum deviation	78.2	57.6
	Order satisfaction rate	11.33%	15.67%
40 jobs	Average deviation	129.8	62.05
	Maximum deviation	152.56	41.55
	Order satisfaction rate	8%	11.75%

Table 2 Comparison of the heuristics with the practical solution with different re-entrant jobs

Problem size	With Re-entrant job no.	Performance measure	Practical	Proposed
20 jobs	1 jobs	Average deviation of makespan	29.28	32.76
		Order on-time rate	20%	21%
	2 jobs	Average deviation of makespan	27.96	34.84
		Order on-time rate	19%	21%
30 jobs	1 jobs	Average deviation of makespan	65.72	47.26
		Order on-time rate	10.60%	13.67%
	2 jobs	Average deviation of makespan	67.68	35.03
		Order on-time rate	9.33%	14.67%
	3 jobs	Average deviation of makespan	74.68	41.67
		Order on-time rate	10%	12.67%
40 jobs	1 jobs	Average deviation of makespan	49.25	38.75
		Order on-time rate	7.50%	11.25%
	2 jobs	Average deviation of makespan	70.48	46.35
		Order on-time rate	8%	9.50%
	3 jobs	Average deviation of makespan	74.917	46.15
		Order on-time rate	7.50%	10.75%
	4 jobs	Average deviation of makespan	94.667	56.75
		Order on-time rate	9%	10.25%

Table 3 Comparison of the heuristics with the practical solution with different the re-entrant job probability

Problem size	**Re-entrant no.**	**Percentage**	**Order satisfaction rate**	
			With re-entrant	Without re-entrant
20 jobs	1 jobs	3%	22.900%	23%
	2 jobs	2%		
30 jobs	1 jobs	3%	15.5700%	15.67%
	2 jobs	1%		
	3 jobs	1%		
40 jobs	1 jobs	3%	11.935500%	11.75%
	2 jobs	1%		
	3 jobs	0.60%		
	4 jobs	0.40%		

One-point crossover

Parent	1010|001110	0011|010010
Offspring	0101|010010	0011|001110

Two-point crossover

Parent	1010|0011|10	0011|0100|10
Offspring	1010|0100|10	0011|0011|10

Uniform crossover

Parent	101|0|001|1|10	001|1|010|0|10
Offspring	101|1|001|0|10	001|0|010|1|10

Fig.1.Crossover operators

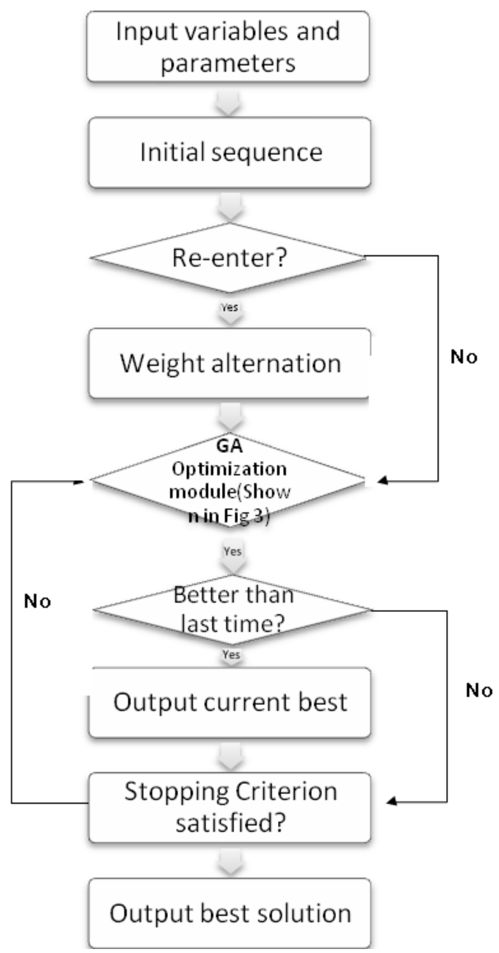

Fig.2. Framework of the optimization of flow shop scheduling.

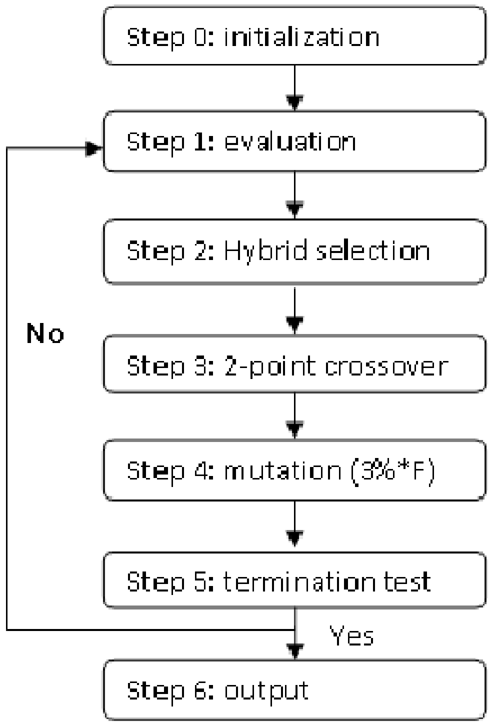

Fig.3. Framework of GA optimization.