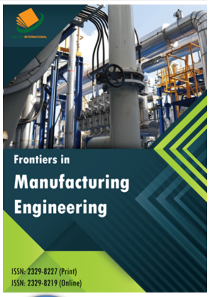
ZIBELINE INTERNATIONAL PUBLISHING

Frontiers in Manufacturing Engineering (FME)

DOI : http://doi.org/10.26480/fme.01.2018.04.07

ISSN: 2329-8227 (Print)
ISSN: 2329-8219 (Online)

CODEN : FMERA8



CrossMark

A HYBRID GENETIC ALGORITHM TO SOLVE MULTI-OBJECTIVE FUZZY FLEXIBLE JOB SHOP SCHEDULING PROBLEM

Kuan Liu*, Hui Cheng, Xinyun Zhang

Shanghai Aerospace Equipments Manufacturer, Shanghai 200000, China
****Corresponding Author E-mail: shlk0324@163.com***

ARTICLE DETAILS

Article History:

Received 12 November 2017
Accepted 10 December 2017
Available online 06 January 2018

ABSTRACT

In this paper, a hybrid genetic algorithm is introduced to overcome the toughest combinatorial optimization problem, the fuzzy-flexible job-shop scheduling. In consideration of the multi-product-and- small-batch production characteristic in aerospace equipments manufacturers, a framework based on hybrid genetic algorithm aimed at minimizing the max makespan and tardiness of workpieces is built to solve the fuzzy flexible job shop scheduling problem. The logistic chaotic mapping model and heuristic rules are introduced in this hybrid genetic algorithm which separates the mutation operation from crossover operation in order to prevent the local optimum happening. This algorithm employs the single point crossover method which protecting the order of procedures to ensure the convergence precision. Meanwhile, experiments are designed to demonstrate the efficiency and feasibility of the Improved Chaotic Genetic Algorithm.

KEYWORDS

Job shop scheduling, multi-objective, genetic algorithm, logistic chaotic mapping model

1. INTRODUCTION

Johnson first proposed the classical Johnson rules to solve n jobs and 2 machines scheduling problem in 1954, establishing the foundation of classical scheduling theory. Since then, lots of scholars pay attention to scheduling problems, especially in the manufacture area. Job Shop Scheduling Problem (JSSP) is one of the most intensive combinatorial optimization problems, which is a typical NP-hard problem. There exists a lot of uncertain factors in scheduling process, with the feature of discreteness, dynamism, multi-variable and constrains, making it the bottleneck problem in the actual production.

The classical JSP consists of scheduling a set of jobs on a set of machines, subject to the constraint that each job has a specified processing order throughout. So, different scheduling rules, heuristic and meta- heuristic algorithms are used to solve JSSP.A genetic algorithm and a scatter search procedure to solve the JSSP is presented by a researcher [1]. According to a researcher, a universal model, which can solve all the problem of SMS (Single Machine Scheduling), PMS (Parallel Machine Scheduling) and FSS (Flow Shop Scheduling), using multi-object Simulated Annealing (SA) [2]. Based on a research, a neural network model focused on detailed scheduling to alternate simulation modeling approach which is costly and time-consuming [3].

The models of classical JSSP mostly assume that all of the processing parameters are known exactly before which is obviously not a realistic approach in manufacturing industry, such as durations, due dates, etc. However, the processing time is uncertain and choosing machine is flexible in real word. Therefore, the actual scheduling problem is a FFJSSP (Fuzzy Flexible Job Shop Scheduling Problem) which releases constrains during the process. Many papers have been published recently. Study showed combine GA (Genetic Algorithm) and VND (Variable Neighborhood Descent) to form a hybrid search algorithm, trying to solve FJSSP (Flexible Job Shop Scheduling Problem) [4]. In literature, an algorithm integrated different strategies is proposed, which is applied in generating the initial population, selecting the individuals and reproducing new individuals [5].

Nevertheless, the combination of flexible and fuzzy JSSP is rarely observed. In this paper, LCS (logistic Chaotic Sequence) is introduced into GA to improve crossover and mutation operation. Considering about the machine tool selection flexibility, a fuzzy flexible job shop scheduling model is constructed with TFN (Triangular Fuzzy Number) to characterize the processing time.

This paper is organized as follows. In section 2, the fuzzy flexible job shop scheduling problem model is described. Section 3 presents an improved hybrid algorithm combined standard GA, LCS with heuristic rules. The simulation tests are illustrated and analyzed in section 4. Finally, a conclusion and directions for future study is covered in Section 5.

2. PROBLEM DESCRIPTION

The N*m FFJSSP can be described as n jobs ($J1, J2, \ldots, Jn$) processed on m machines ($M1$, $M2, \ldots, Mm$), and each job J_i contains h_i sequenced procedure. The j-th working procedure of i-th job is denoted as Oij which can be processed on a set of machines Mij and Oij should be assigned to one of these machines. If Oij can be processed on any machine of group M, that is, $Mij = M$, we call it total flexible JSSP, otherwise partial flexible JSSP.

Let a triangle fuzzy number $A_{ijk} = (d^1_{ijk}, d^2_{ijk}, d^3_{ijk})$ represents processing time of procedure Oij on machine M_k. The triangular membership function can be expressed as:

$$u_{ijk}(x) = \begin{cases} 0, & x \le d^1_{ijk} \\ \dfrac{x - d^1_{ijk}}{d^2_{ijk} - d^1_{ijk}}, & d^1_{ijk} < x \le d^2_{ijk} \\ \dfrac{d^3_{ijk} - x}{d^3_{ijk} - d^2_{ijk}}, & d^2_{ijk} < x \le d^3_{ijk} \\ 0, & x > d^3_{ijk} \end{cases} \tag{1}$$

Wherein d^1_{ijk} denotes the shortest processing time; d^2_{ijk} means the most possible processing time; d^3_{ijk} is the longest processing time. Accordingly, let the triangular fuzzy number S_{ijk} and E_{ijk} denote the start time and the completion time of O_{ij} respectively. Hence, we have $E_{ijk} = S_{ijk} + A_{ijk}$.

Actually, not every workpiece can be finished on time, so we have to evaluate the satisfaction on the finishing time. Thus, the satisfaction degree of E_i (the completion time of job i) can be measured by a linear proportion.

Study showed, the satisfaction degree D_i [6]:

$$D_i = \frac{area(E_i \cap U_i)}{area E_i} \tag{2}$$

Where, U_i is the due date of J_i.

This paper adopts the multi-objective optimization, to minimize the maximum completion and the tardiness of workpieces.

The objective function is:

$$\begin{cases} F_1 = \min E = \max\limits_{i=1:n}(\max\limits_{j=1:h_i}(\max\limits_{k=1:m} E_{ijk}) \\ F_2 = \min(\text{tardiness}) = \max\limits_{i=1:n} D_i \end{cases} \tag{3}$$

The constraint function is:

$$\begin{cases} E_{ij} \le S_{i(j+1)} & 1 \le i, q \le n \\ E_{ij} \le S_{pq} \vee E_{pq} \le S_{ij} & 1 \le j, p \le h_i \\ S_{ij} \ge 0 & i, j \in R^+ \end{cases} \tag{4}$$

Constraint definition:

(1) The workpieces must be carried out in accordance with the established procedures;

(2) Any machine at one time can only process one procedure, and cannot be interrupted in processing;

(3) Any machine can immediately be putting into use at the beginning of time zero.

3. THE HYBRID GENETIC ALGORITHM

The classical genetic algorithm generates the initial population randomly, which will increase the diversity of the solution space. However, it will undoubtedly slow down the rate of convergence. At the same time, the mutation operation is based on cros operation commonly. The classical genetic algorithm only emphasizes the importance of the cross operation in the evolution process and ignore the contribution of mutation.

Hence, this paper proposes to generate the initial population by using a random method combing with heuristic rules. Meanwhile, the mutation operation is separated from the cross operation, making it the independent optimization operation paralleled to the cross operation. In addition, the introduction of Logistic chaotic mapping model determining the intersection point with the chaotic sequence employing the single point crossover method ensures the convergence precision. In mutation operation, we use the chaotic sequence to mutate a variety of chromosome in order to avoid the premature distress of this algorithm.

3.1 Encoding and Decoding Method

Define abbreviations and acronyms the first time they are used in the text, If FFJSSP uses the traditional JSSP procedure singular encoding method, the factor of machine flexibility will be ignored. In order to overcome this drawback, the double chromosome encoding method based on the decimal coding is adopted that one chromosome represents the sequence of procedure and another represents the distribution of machines. Of course, the two chromosomes correspond to each other one by one according to the position, as shown in Table 1.

Table 1: The method of encoding and decoding.

Procedure encoding	1	3	1	2	2	3
Procedure decoding	O_{11}	O_{31}	O_{12}	O_{21}	O_{22}	O_{32}
Assignment encoding	1	2	3	2	3	1
Assignment decoding	M_2	M_2	M_3	M_2	M_3	M_1

The genes of the procedure chromosome denote the workpiece, and the order in which they appear describes the sequence of operations. So is same with the machine assignment encoding method.

Table 2: The method of encoding and decoding.

	M_1	M_2	M_3	M_1	M_2	M_3		M_1	M_2	M_3
O_{11}	2	3	1	2	3	①	…	2	3	①
O_{12}	2	3	2	2	3	*3*		②	3	2
O_{21}	4	3	5	4	3	*6*		4	③	5
O_{22}	4	3	5	4	3	*6*		④	3	5

3.2 Initial Population

The initial population is generated by using the approach by localization of a researcher [7]. As aforementioned, the strategy of balancing the average processing time of each machine is used, i.e. to find the local minimum processing time machine according to the procedure chromosome generated randomly.

As shown in Table 2, assuming a random processing sequence $\{O_{11}O_{21}O_{12}O_{22}\}$, the machine with minimum processing time corresponding to procedure O_{11} is selected firstly and then the processing time is accumulated to the other procedures on this machine (italic values in Table 2 express the machines workload updating constantly). Do this operation until the last procedure is assigned.

3.3 Fitness Function

As mentioned before, the completion time is expressed by the TFN. If we use the TFNs in the fitness function, the calculating data are very cumbersome in the genetic algorithm iteration operation. It's convenient to use the expected value method to get the expectation of the completion time, expressed as follows:

$$T = \beta \int_{d^1_{ijk}}^{d^2_{ijk}} \frac{x - d^1_{ijk}}{d^2_{ijk} - d^1_{ijk}} + (1-\beta) \int_{d^2_{ijk}}^{d^3_{ijk}} \frac{d^3_{ijk} - x}{d^3_{ijk} - d^2_{ijk}} \tag{5}$$

Because the objective is to minimize the completion time and maximize the satisfaction degree of all the workpieces, the fitness function can be constructed like Equation (6).

$$\begin{cases} Fitness1 = \max(1/T) \\ Fitness2 = \max(D) \end{cases} \tag{6}$$

3.4 Selection Operation

In this study, we chose the classical roulette wheel mechanism [8]. The individual fitness value is obtained from Equation (3), and the selection probability of each chromosome is generated based on Equation (7).

$$\Delta P_i = Fitness(i) \Big/ \sum_{i=1}^{n} Fitness(i) \tag{7}$$

Meanwhile, the probability interval that each individual occupied is cumulated from zero to 1, that is to say, $P_{i+1} = P_i + \Delta P_{i+1}$ ($P_1 = \Delta P_1$). Obviously, the individual is more likely to be chosen while its probability interval is larger. Then the selection operation is to choose the individuals that the numbers generated randomly ranging from 0 to 1 belong to their probability intervals.

3.5 Crossover Operation

Two rules are taken into consideration while the parents are arranged in pairs before applying the crossover operation. Rule 1 indicates the elitism strategy and it also may bring the premature feasibility. Rule 2 is a common method.

Rule 1: match each other according to the fitness value sort order (MF for shorten).

Rule 2: match each other randomly (MR for shorten).

In crossover operation, the cross points are generated by using the logistic chaotic sequence. The process of crossover operation is shown as follows:

Generate a pseudo random number between 0 and 1 as the initial value.

Use the logistic chaotic sequence x(n+1)=4x(n)[1-x(n)] to generate a chaotic number and save this value as the next iteration initial value.

Multiply the value in 2) by the length of chromosome to get the cross point number.

Keep all the parent genes before the cross point, and insert all the genes after the cross point which doesn't appear in parent 1 following the order in parent 2 to generate child 1, the child 2 generated similarly (See in Figure 1.)

Machine assignment crossover operation corresponding to the procedure chromosome crossover operation one by one.

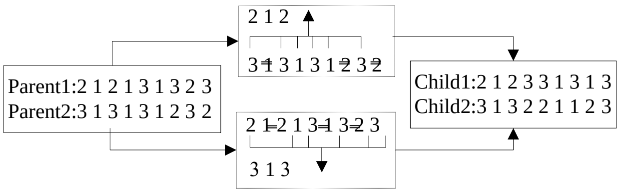


Figure 1: Crossover operation based on preserving the procedure order.

3.6 Mutation Operation

Mutation operation is a crucial step in the genetic algorithm, determining the algorithm performance. Weather can the algorithm jump out of local optimum trap to search the global optimization or not depends on the mutation operation set. Improved mutation operation adopted in this paper is separated from the crossover operation, selecting two mutation genes from chromosome using logistic chaotic sequence and exchanging with each other to get a new chromosome.

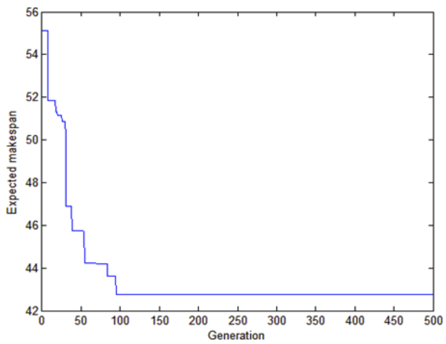


Figure 2: Convergence curve.

4. EXPERIMENTS

It's hard to find the benchmarks of FFJSSP, so we choose the processing data from the actual processing manufacturer in this section. A 10×10 (50procedures, 10 machines) scheduling problem example is presented. Let D=[50, 60, 70, 40, 60, 60, 60, 60, 60, 65] denote due date of each jobs.

We use MATLAB to simulate the classical genetic algorithm (CGA) and the improved chaos genetic algorithm (ICGA). The parameters are set as follows: population size=500; times of iteration=200; β=0.7; the crossover probability Pc=0.8; the mutation probability Pm=0.05. Run 10 times, the convergence curve is shown in Figure 2.

Table 3: Comparison of results using cga and icga.

	1	2	3	4	5
CGA	58.70	53.55	53.75	51.85	54.65
ICGA	41.85	40.95	39.75	41.25	42.20
	6	7	8	9	10
CGA	57.10	57.75	55.90	56.25	54.85
ICGA	40.35	43.40	42.25	43.00	42.00

In the 10 operations, the optimal SGA value is 51.85, the average value was 55.44; the optimal ICGA algorithm is proposed in this paper is 39.75, the average value is 41.7 (as shown in Table 3).

Obviously, the improved genetic algorithm proposed in this paper has a better robustness, and the results of ICGA are close to the optimal one. Using ICGA can shorten the makespan about 25% than CGA. Obviously, the algorithm we proposed has a better performance in optimization effect than CGA.

The best operation with shorter completion times E_i and the satisfaction degree D_i of the ten jobs are shown in Table 4. We use a Gantt chart in Figure 3. to show the job shop scheduling intuitively, in which the triangles above machine line represent the fuzzy completion time and the below ones stand for the fuzzy start time of each procedures. From the fuzzy scheduling Gantt chart, the improved genetic algorithm proposed in this paper obtained a relatively balanced distribution scheduling scheme of machines. Using this method, the machine utilization rate is higher, and the completion time is shorter.

Table 4: The completion time and satisfaction degree.

Job	1	2	3	4	5	6	7	8	9	10
E_i	(31,46,60)	(34,50,66)	(38,55,74)	(18,25,32)	(30,36,47)	(39,54,75)	(38,53,73)	(37,53,72)	(40,57,73)	(40,56,72)
D_i	0.7537	0.9297	0.9766	1	1	0.7024	0.7586	0.7835	0.6586	0.9043

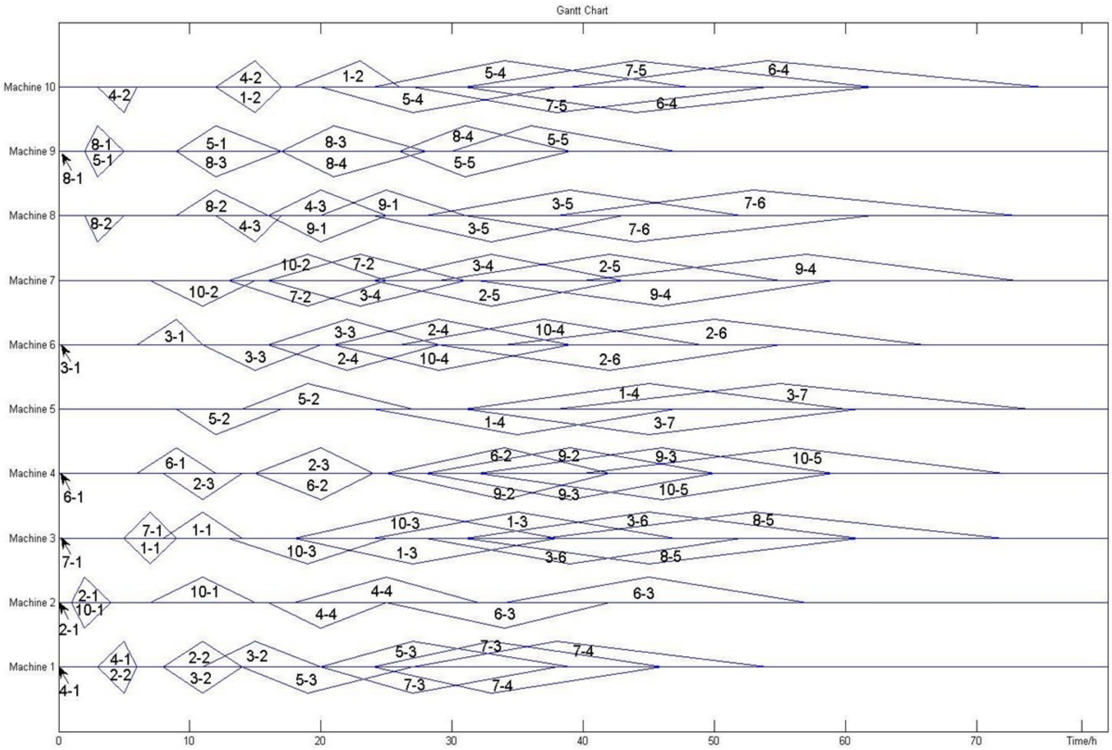


Figure 3: Gantt chart of scheduling.

5. CONCLUSIONS

We have developed a new approach hybridizing heuristic rules with genetic algorithm to form an effective method to solve the flexible fuzzy job shop scheduling problem. The chaos theory is introduced into our hybrid genetic algorithm. The mutation operation is based on the cross operation in the classical structure of the genetic algorithm. The improved genetic algorithm adopted in this paper highlights the importance of mutation operation, which is separated from the crossover operation to guide the algorithm out of local optimum. The crossover operation uses the single point crossover based on preserving the procedure order to avoid the chattering phenomenon to ensure the accuracy of the algorithm. Meanwhile, we choose multi chromosome Mutation to make up for the probability that single point crossover operation get into the premature condition. From the Gantt chart of scheduling, it can be directly found that although the improved algorithm can get a better scheduling, but there still exist some bottleneck machine. Our future research direction is to find some methods to insert procedures into the idle time of machines to solve the bottleneck machine problem, which makes the machines distributed more balanced and improves the efficiency of the manufacturers.

REFERENCES

[1] Sels, V., Craeymeersch, K., Vanhoucke, M. 2011. A hybrid single and dual population search procedure for the job shop scheduling problem, European Journal of Operational Research, 215, 512-523.

[2] Loukil, T., Teghem, J., Tuyttens, D. 2005.Solving multi- objective production scheduling problems using metaheuristics, European journal of operational research, 161, (1), 42-61.

[3] Golmohammadi, D. 2013. A neural network decision-making for job-shop scheduling, International Journal of Production Research, 51, (17), 5142-5157.

[4] Gao, J., Sun, L., Gen, M. 2008. A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems, Computers and Operations Research, 35, (9), 2892-2907.

[5] Pezzella, F., Morganti, G., Ciaschetti, G. 2008. A genetic algorithm for the Flexible Job-shop Scheduling Problem, Computers and Operations Research, 35, 3202–3212, 2008.

[6] Gonzales-Rodriguez, I., Puente, J., Vela, C.R., Varela, R. 2008. Semantics of Schedules for the Fuzzy Job-Shop Problem, IEEE Transactions on Systems, Man, and Cybernetics, Part A, 38, (3), 655-665.

[7] Kacem, I., Hammadi, S., Borne, P. 2002. Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems, IEEE Transactions on Systems, Man, and Cybernetics, Part C, 32, (1), 1-13.

[8] Liu, J.J. 2009. Application of optimization Genetic Algorithm in Fuzzy Job Shop Scheduling Problem, IEEE Global Congress on Intelligent Systems, 1, 436-440.