

An Intelligent Method for Solving Flexible Job Shop Scheduling Problem

Qiaofeng Meng¹

National CIMS Engineering Technology Research Center at Tsinghua University

Beijing, 100084, China

E-mail: wmmymmqff@163.com

Linxuan Zhang

National CIMS Engineering Technology Research Center at Tsinghua University

Beijing, 100084, China

E-mail: lxzhang@mail.tsinghua.edu.cn

Yushun Fan

National CIMS Engineering Technology Research Center at Tsinghua University

Beijing, 100084, China

E-mail: fanyus@tsinghua.edu.cn

Haiwei Luo^a, Hongjie Zhao^b

Capital Aerospace Machinery Company

Beijing, 100076, China

E-mail: ^amqf12@mails.tsinghua.edu.cn; ^b1343854830@qq.com

An efficient intelligent algorithm is proposed in this paper to solve the flexible job shop scheduling problem (FJSSP). The algorithm takes genetic algorithm (GA) as the main frame, and combines tabu search (TS) with simulated annealing (SA) to promote local search. The operation-based representation, novel crossover and mutation operation are introduced to increase individual diversity. When the current solution exists in the tabu list, the algorithm performs local search incorporating TS and the SA method to explore the neighbourhood of the individuals. The experiments are carried out on 10 different scales instances and the results with other algorithms are compared. The results conclude that the proposed algorithm has advantages on both the quality and the time consuming which is effective to the actual FJSSP.

CENet2017

22-23 July, 2017

Shanghai, China

¹Speaker

The work was supported by the National Science and Technology Supporting Plan of China No. 2012BAF15G01).

Corresponding Author

1. Introduction

If the constraints of the job shop scheduling process are relaxed, the problem becomes a flexible job shop scheduling problem (FJSSP) [1,2]. Due to the complexity of FJSSP, it is not easy to obtain the ideal solutions in finite time with the classical mathematical method. While meta-heuristic algorithm does not necessarily establish accurate mathematical models of the problem. We can converge gradually to a near optimal solution via evolution. The advantages make the meta-heuristics algorithm become the main method to solve the FJSSP, which has been widely concerned by scholars in recent years[3-7].

Pezzella F, et al.[8] used GA method to solve the scheduling problem and proposed an initial value generation method. Tay J C et al.[9] proposed a new method of individual representation in population. Vaghefinezhad S. et al.[10] applied GA to solve FJSSP with time-varying demands. Zhang G et al.[11] put up with a GA algorithm for optimizing the makespan of FJSSP. Driss I, et al.[12] redesigned the crossover and mutation operations for FJSSP. Zhou W et al. [13] introduced a hybrid GA and SA algorithm. Li X, et al. [14] designed an algorithm that combines GA and TS to solve FJSSP. Demir and Isleyen (2014) [15] put forward a method of GA to solve FJSSP with the cross process. Moghadam A M et al.[16] devised a new GA for FJSSP to minimize the makespan.

From the above review, scholars have done some algorithm researches on the FJSSP problem in the past few decades. However, most of the algorithms can't have both good global performance and local search performances simultaneously. A hybrid GA algorithm that takes both global and local search capabilities for FJSSP into account is proposed in the paper.

The contents of this paper are arranged as follows. Following the introduction, the next section defines the mathematical model. In Section 3, the algorithm proposed in this paper is discussed. In Section 4, the effectiveness of the algorithm is verified by experiments. Finally, we sum up the contents of this paper and propose the future research plan.

2.The Mathematical Model for FJSSP

The mathematical model of FJSSP is established as follows.

$$F = \min \{ \max(C_{ij} \mid 1 \leq i \leq N_i, 1 \leq j \leq n) \} \quad (2.1)$$

s.t.

$$B_{ij} \geq 0 \quad (2.2)$$

$$C_{ij} = B_{ij} + P_{ij} \quad (2.3)$$

$$B_{i,j+1} \geq C_{ij} \quad (2.4)$$

$$x_{tik} = 1 (B_{ij} \leq t \leq C_{ij}) \quad (2.5)$$

$$x_{ia} \cdot x_{ib} = 0 \quad (a \neq b) \quad (2.6)$$

$$x_{ick} \cdot x_{idk} = 0 \quad (c \neq d) \quad (2.7)$$

$$\sum_{k=1}^m y_{kij} \geq 1 \quad | \{ M_k \mid y_{kij} \neq 0, 1 \leq k \leq m, 1 \leq j \leq N_i \} | \leq N_i \quad (2.8)$$

$$\{ M_k \mid y_{kij} \neq 0, 1 \leq k \leq m, 1 \leq j \leq N_i \} \in \{ M_1, M_2, \dots, M_k \}$$

$$x_{tik} = \begin{cases} 1, & \text{job } J_i \text{ is handled on the } k\text{th machine at } t \\ 0, & \text{others} \end{cases}$$

$$y_{kij} = \begin{cases} 1, & O_{ij} \text{ can be handled on the } k\text{th machine} \\ 0, & \text{others} \end{cases}$$

Where N_i is the number of operations for job J_i , B_{ij} is start time of O_{ij} , C_{ij} is the completion time for O_{ij} , x_{tik} is the decision variable for that job J_i processed on the machine k within time t , y_{kij} is the decision variable that O_{ij} handled on the k th machine.

The above formula (2.1) defines the equation of the objective which is to minimize makespan. The formula (2.2) indicates that all jobs are processed after the start time of scheduling. The formula (2.3) defines that when a job is operated on a machine it can't be interrupted. The formula (2.4) defines that each operation must be in accordance with the sequence of processing, that is, the current operation can't be processed until previous operation (if it exists) finishes. The formula (2.5) defines that if it is not interrupted during the process the machine has the ability to process without considering the fault and maintenance. The formula (2.6) means that the same job can't be processed on two machines at the same time. And the formula (2.7) indicates that the same machine can't handle two jobs simultaneously. The inequality (2.8) concludes that each operation has multiple optional machines, in which $|\cdot|$ is used for calculating the number of elements.

3.The Proposed Hybrid Genetic Algorithm

The meta-heuristic algorithm which can effectively converge to the optimal solution to solve FJSSP is proved to be the key factor [17]. A new hybrid algorithm (NHGA) which combines GA, TS and SA is proposed. The algorithm not only has good global searching performance, but also has excellent local searching capability. The GA which has powerful global searching ability is utilized to search in global space[18,19], and TS combined with SA which has good local searching ability is applied to search in local space[20,21]. The main steps of the algorithm are described as follows.

- Step 1: Initialize all the parameters for the proposed NHGA;
- Step 2: Generate the initial search group randomly;
- Step 3: Compute the objective function;
- Step 4: Designate the fitness value to each individual based on the objective function;
- Step 5: Execute selection operations;
- Step 6: Perform crossover operations;
- Step 7: Run mutation operations;
- Step 8: Carry out local search with SA and TS;
- Step 9: Finishes the algorithms if the solution satisfies the condition, otherwise go to Step

3.

3.1 Chromosome Representation

Chromosomes are related to the decryption of FJSSP. Since FJSSP contains two sub-problems, there are two parts to encode correspondingly: process sequence and machine allocation. The method for encoding in Pezzella F et al. [4] is utilized in the paper.

3.2 Local Search with SA and TS

If the current solution is in the tabu list and has emerged several times, it is likely to be a local optimal value. In order to make the algorithm jump out of the local optimal value, local search is performed.

$$\Delta f = f(s') - f(s) \quad (3.1)$$

Where s is current solution, and s' is the neighbourhood of s .

The key step is depicted as bellow.

Step 1: Set the initial annealing temperature;

Step 2: Generate a feasible solution around the current solution. Calculate the difference according to formula (3.1). Decide whether to accept the solution according to some probability condition;

Step 3: Decrease the temperature ;

Step 4: If the temperature is greater than the termination temperature, then jump out of the local search, or go to step 2.

4. Simulation

The NHGA algorithm proposed is written in Matlab and run on a 4.0 GHz PC and 8G memory computer.

To evaluate the efficiency, the NHGA algorithm is tested on dataset of Brandimarte [2]. The results of NHGA are compared with the results of GA proposed by F. Pezzella et al. [8], eGA is proposed by Moghadam A M et al. [16], HA proposed by Li X et al. [14] and NGA proposed by I. Driss et al. [12]. The deviation of the best results between NHGA and other algorithms is computed by Dev , as shown in Formula (4.1).

$$Dev = [(F_{NHGA} - F_{BEST}) / F_{BEST}] \times 100\% \quad (4.1)$$

Where Dev is the relative deviation, F_{NHGA} is the best makespan of NHGA and F_{BEST} is the best makespan of other algorithms.

To obtain meaningful makespan, the proposed algorithm was run five times on the same condition. Table 1 summarizes the experimental results. In each experiment, the NHGA algorithm achieves the best makespan than any other algorithms among all 10 instances. And in some instances, the algorithm refreshes better solution such as MK03, MK04 and MK06. Table 2 compares the computation time of the NHGA algorithm on each instance with those of other algorithms. In addition, the NHGA algorithm spends less time on other instances than other algorithms. Therefore, the NHGA algorithm is effectively used to solve FJSSP.

Instances	Size	LB	GA	eGA	HA	NGA	HNGA	Dev(%)
MK01	(10,6)	36	40	40	40	37	37	0
MK02	(10,6)	24	26	26	26	26	26	0
MK03	(15,8)	204	204	204	204	204	203	-0.49
MK04	(15,8)	48	60	60	60	60	59	-1.7
MK05	(15,4)	168	173	173	172	173	172	0
MK06	(10,15)	33	63	58	57	67	56	-1.75
MK07	(20,5)	133	139	144	139	148	139	0
MK08	(20,10)	523	523	523	523	523	523	0
MK09	(20,10)	299	311	307	307	307	307	0
MK10	(20,15)	165	212	198	197	198	197	0

Table 1: Comparison Between NHGA and Other Algorithms in Minimizing Completion Time

Instances	Size	GA	eGA	HA	NGA	HNGA	Dev(%)
MK01	(10,6)	1.8	1.6	0.06	2	0.08	33
MK02	(10,6)	3	2.6	0.59	4	0.5	-15.3
MK03	(15,8)	2	1.3	0.16	1.8	0.14	-12.5
MK04	(15,8)	8	6.2	0.49	10	0.42	-14.3
MK05	(15,4)	7	7.3	4.57	9	4.4	-3.7
MK06	(10,15)	12	15.7	53.82	15	12	-6.7
MK07	(20,5)	15	17.3	20.01	17	15	0
MK08	(20,10)	10	2.2	1.8	11	2	11
MK09	(20,10)	18	30.2	2.6	17	3	15.4
MK10	(20,15)	38	36.6	33.21	35	30	-9.6

Table 2: Comparison Between NHGA and Other Algorithms in Computing Time

From the above calculation, we conclude that the NHGA not only has the advantage in the quality of the solution, but also has more advantages over the other algorithms in time consuming. The main reasons are as follows. Firstly, the population can remain the excellent gene and converge to the optimal solution with the favorable ability of the global search of GA. Furthermore, this paper uses SA and TS for itself, which greatly improves the efficiency of the algorithm. For large-scale scheduling problems, the proposed algorithm can also achieve optimal makespan in a limited amount of time. The reason it that the algorithm effectively achieves the balance of exploitation and exploration.

5. Conclusion

This paper presents an efficient hybrid algorithm to FJSSP, whose objective is to optimize makespan. The algorithm uses GA to produce an initial solution set and then uses TS and SA for local search. In this way, the good global search performance of GA is successfully integrated with the excellent local search performance of TS and SA for FJSSP, which can balance the intensification and diversification very well. The validity of the algorithm is verified by experiments, and the experimental results of the NHGA algorithm are compared with those of other popular algorithms. Simulation results show that the NHGA algorithm has great advantages in both computation and time consumption. Based on the paper research, the next research project is defined as: conducting more experiments on the parameters of the algorithm, which can improve the performance efficiently and Applying NHGA to solve the multi-objective FJSSP.

References

- [1] MR Garey, DS Johnson, R Sethi, *The Complexity of Flowshop and Jobshop Scheduling*[J]. *Mathematics of Operations Research*,1976, 1(2):117-129.

- [2] Brandimarte, Paolo. *Routing and scheduling in a flexible job shop by tabu search*[J]. Annals of Operations Research, 1993, 41(3):157-183.
- [3] Jia S, Hu Z H. *Path-relinking Tabu search for the multi-objective flexible job shop scheduling problem*[J]. Computers & Operations Research, 2014, 47(9):11–26.
- [4] Shahsavari-Pour N, Ghasemishabankareh B. *A novel hybrid meta-heuristic algorithm for solving multi objective flexible job shop scheduling*[J]. Journal of Manufacturing Systems, 2013, 32(4):771–780.
- [5] Rossi A. *Flexible job shop scheduling with sequence-dependent setup and transportation times by ant colony with reinforced pheromone relationships*[J]. International Journal of Production Economics, 2014, 153(4):253-267.
- [6] Gao C, Gao Y, Lv S. *Improved simulated annealing algorithm for flexible job shop scheduling problems*[C]// Chinese Control and Decision Conference. IEEE, 2016.
- [7] Shen X N, Yao X. *Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems*[J]. Information Sciences, 2015, 298:198-224.
- [8] Pezzella F, Morganti G, Ciaschetti G. *A genetic algorithm for the Flexible Job-shop Scheduling Problem*[J]. Computers & Operations Research, 2008, 35(10):3202-3212.
- [9] Tay J C, Wibowo D. *An Effective Chromosome Representation for Evolving Flexible Job Shop Schedules*[M]// Genetic and Evolutionary Computation – GECCO 2004. Springer Berlin Heidelberg, 2004:210-221.
- [10] Vaghefinezhad S, Wong K Y. *A Genetic Algorithm Approach for Solving a Flexible Job Shop Scheduling Problem*[J]. International Journal of Computer Science Issues, 2012, 9(3).
- [11] Zhang G, Gao L, Shi Y. *An effective genetic algorithm for the flexible job-shop scheduling problem*[J]. 2011, 38(4):3563-3573.
- [12] Driss I, Mouss K N, Laggoun A. *A new genetic algorithm for flexible job-shop scheduling problems*[J]. Journal of Mechanical Science and Technology, 2015, 29(3):1273-1281.
- [13] Zhou W, Bu Y P, Zhou Y Q. *An Improved Genetic Algorithm for Solving Flexible Job shop Scheduling Problem*[C]// Control and Decision Conference. 2013:4553-4558.
- [14] Li X, Gao L. *An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem*[J]. International Journal of Production Economics, 2016, 174:93-110.
- [15] Demir Y, İşleyen S K. *An effective genetic algorithm for flexible job-shop scheduling with overlapping in operations*[J]. International Journal of Production Research, 2014, 52(13):3905-3921.
- [16] Moghadam A M, Wong K Y, Piroozfard H. *An efficient genetic algorithm for flexible job-shop scheduling problem*[C]// IEEE International Conference on Industrial Engineering and Engineering Management. IEEE, 2014:1409-1413.
- [17] Ling Wang, Gang Zhou, Ye Xu, et al. *A hybrid artificial bee colony algorithm for the fuzzy flexible job-shop scheduling problem*[J]. International Journal of Production Research, 2013, 51(12):3593-3608.
- [18] Man Z, Wei T, Xiang L, et al. *Research on Multi-project Scheduling Problem Based on Hybrid Genetic Algorithm*[C]// International Conference on Computer Science and Software Engineering. IEEE, 2008:390-394.
- [19] Wang J F, Du B Q, Ding H M. *A Genetic Algorithm for the Flexible Job-Shop Scheduling Problem*[M]// Advanced Research on Computer Science and Information Engineering. Springer Berlin Heidelberg, 2011:3202-3212.

- [20] Hurink J, Jurisch B, Thole M. *Tabu search for the job-shop scheduling problem with multi-purpose machines*[J]. OR Spectrum, 1994, 15(4):205-215.
- [21] Billaut J C. *A tabu search algorithm for solving a multicriteria flexible job shop scheduling problem*[J]. International Journal of Production Research, 2011, 49(23):6963-6980.

POS (CENet2017) 050