

# An Improved Ant Colony Optimization for Flexible Job Shop Scheduling Problem

Lei WANG<sup>1\*</sup>, Jingcao CAI<sup>2</sup>, Zhihu LIU<sup>3</sup>

School of Mechanical and Automotive Engineering  
Anhui Polytechnic University  
Wuhu, 241000, China

<sup>1</sup>\*E-mail: wangdalei2000@126.com ; <sup>2</sup>E-mail:  
390096598@qq.com ; <sup>3</sup>E-mail: 895616635@qq.com

+\* Corresponding author

Chaomin LUO

Department of Electrical and Computer Engineering  
University of Detroit Mercy  
Detroit, 48221, USA  
E-mail: luoch@udmercy.edu

**Abstract**—This paper deals with the flexible job shop scheduling problem (FJSP) with the objective of minimizing the makespan. An improved ant colony optimization was proposed to avoid falling into the local optimum and time consuming exist in the basic ant colony optimization (ACO). Computational results show that the proposed improved ant colony optimization algorithm can obtain effective solutions in very short and nearly zero time and is comparable with some other heuristic algorithms and can effectively solve the FJSP.

**Keywords**- flexible job shop scheduling problem; makespan; improved ant colony optimization

## I. INTRODUCTION

In recent years, scheduling plays a vital role in manufacturing environments due to the growing consumer demand for variety, reduced product life cycles, changing markets with global competition, and the rapid development of new technologies [1].

The job shop scheduling problem (JSP) is one of the most popular scheduling problems existing in practice. It has attracted the attention of many researchers [2-4] due to its wide applicability and inherent difficulty [1]. the flexible job shop scheduling problem (FJSP) is generally similar to the classical job-shop scheduling problem (JSP), but differs in the details of the machine assignment. In the JSP, each operation can only be performed on one machine out of a set of machines. However, the FJSP is more complex than JSP. FJSP allows each operation to be processed on more than one machine. However, different machines require different processing time for each operation [5]. Thus, FJSP is more complicated than JSP problem and it has been proved that the FJSP is strongly NP-hard in 1993 [6].

In recent years, many researchers have applied meta-heuristic algorithms to solve this problem. meta-heuristic algorithms include simulated annealing (SA), genetic algorithm (GA), tabu search (TS), artificial bee colony (ABC), particle swarm optimization (PSO), ant colony optimization (ACO) and so on.

Kacem et al. [7] proposed an effective evolutionary algorithm. Zhang et al. [8] introduced a hybrid algorithm combining PSO algorithm with TS algorithm. Ho et al. [9] studied a hybrid evolution algorithm combined with a guided local search and an external Pareto archive set. Xing et al.

[10] proposed a local search algorithm for the multi-objective FJSPs. Many researchers also consider the multi-objective FJSPs with new and interesting directions, such as the maintenance activities constraints [11].

In this paper, an improved ant colony optimization was proposed to avoid falling into the local optimum and time consuming exist in the basic ant colony optimization (ACO) algorithm. The results are found to be closer or equal to the global optimum. In addition, the computational time is better than some other algorithms.

## II. DESCRIPTION OF FJSP

The  $n \times m$  FJSP can be defined as follows [17]: There is a set of  $n$  jobs  $J = \{J_1, J_2, \dots, J_n\}$  and a set of  $m$  machines  $M = \{M_1, M_2, \dots, M_m\}$ . Each job  $i$  consists of a sequence of operations  $\{o_{i1}, o_{i2}, o_{i3}, \dots, o_{in_i}\}$  where  $n_i$  is the number of operations that job  $i$  comprises. Each operation  $o_{ik}$  has to be processed by one machine from the machine set  $M_{ij} \subseteq M$ .

The problem is thus to both determine an assignment and a sequence of operations on the machines to satisfy some criteria. It contains of two sub-problems: the machine selection problem and the operation sequencing problem. Therefore, the FJSP is more complicated and challenging than the classical JSP because it requires a proper selection of a machine from a set of available machines to process each operation of every job [9].

The hypotheses considered in this paper are summarized as follows [12]: (1) the different operations of one job cannot be processed simultaneously. (2) Jobs are independent. Job preemption is not permitted and each machine can handle only one job at a time. (3) All jobs and machines are available at time zero simultaneously. (4) Setup time for the operations on the machines is independent of the operation sequence and is incorporated into the processing time. (5) After a job is processed on a machine, it is immediately transported to the next machine on its process, and the transmission time is assumed to be negligible.

The scheduling objective is to minimize the maximal makespan. The mathematical model of FJSP can refer to [13].

### III. IMPROVED ANT COLONY OPTIMIZATION FOR FJSP

The main steps of ACO are the calculations of transition probability, visibility and pheromone amount. Ants choose their next nodes based on pheromone amount and visibility [14]. At time  $t$ , the probability for ant  $k$  choosing the path from node  $i$  to node  $j$  is calculated, as shown in (1).

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{s \in allowed_k} \tau_{is}^\alpha(t) \eta_{is}^\beta(t)}, & \text{if } j \in allowed_k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where  $allowed_k$  is selectable machines group that ant  $k$  can choose,  $\alpha$  and  $\beta$  are the pheromone and expectation factor

respectively,  $\tau_{ij}(t)$  is pheromone amount between node  $i$  and  $j$  at time  $t$ .  $\eta_{ij}(t)$  is heuristic factor and it is computed as  $\eta_{ij}(t) = 1/f(x)$ . The more the pheromone amount on the path is and the higher the visibility is, the bigger probability to choose this path is.

However, ant selects the next available machine by employing the transition probability does not always obtain the optimal direction, and the pheromone deviated from optimal solution has the potential to be enhanced, which will easily lead to the local optimum.

Therefore, a new node selection method which combining prior knowledge, probability search and random search is proposed. When the search is becoming trapped in local optimal solution, the solution space can be further searched by adjusting the pheromone and increasing the random selection probabilities.

$$p = \begin{cases} p_1 = \arg \max_{j \in allowed_k} \{ [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta \cdot \xi_{ij}(t) \} & 0 \leq q \leq q_0 \\ p_2 = p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t) \xi_{ij}(t)}{\sum_{s \in allowed_k} \tau_{is}^\alpha(t) \eta_{is}^\beta(t) \xi_{is}(t)}, & j \in allowed_k \\ 0 & \text{otherwise} \end{cases} & q_0 < q \leq q_1 \\ p_3 = \text{random search} & q_1 < q \leq 1 \end{cases} \quad (2)$$

where  $q$  is a random value and  $0 \leq q < 1$ .  $q_0$  is the degree of prior knowledge,  $q_1$  is the lower bound level of random search,  $\xi_{ij}(t)$  represents the appearing number of arc  $(i, j)$  during previous iterations for producing optimal solution.

The more the arc  $(i, j)$  appears, the more the role plays for searching good solution by using positive feedback.

A good balance relation between "using the past information to speed up the convergence" and "exploring new paths" can be established by combining these three selection methods. By this way, the search space can be enlarged and the global optimum can be obtained with a larger probability.

As time goes on, the pheromone of the path evaporate gradually. Ants travel a cycle after time  $n$ , an ant will modify the amount of pheromone on the passed edges by applying the local updating rule, as shown in (3).

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \rho \tau_0 \quad (3)$$

where  $\rho$  ( $0 < \rho < 1$ ) is the evaporation coefficient of pheromone.  $\tau_0$  is the initial value of pheromone on all edges.

Once all ants have arrived at their destination, the amount of pheromone on the edge is modified again by applying the global updating rule, as shown in (4).

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \rho \Delta \tau_{ij}^k(t) \quad (4)$$

where  $\Delta \tau_{ij}^k(t)$  is the quantity of pheromone on the path  $(i, j)$  laid by ant  $k$ , and it can be defined as:

$$\Delta \tau_{ij}^k(t) = \begin{cases} Q/L^k & \text{if ant } k \text{ travel through arc } (i, j) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

In (5),  $Q$  is a constant and it denotes the strength of pheromone;  $L^k$  is evaluation value of the  $k$ th ant after finish the search task.

### IV. EXPERIMENTAL RESULTS

Table I provides the processing information about 4 jobs and 6 machines for a FJSP problem that is used for illustrating the proposed improved ACO. The numbers in Table I represents processing time on each machine and "-"

represents the operation can not be processed on this machine.

The parameters and their values, which are used for running the improved ACO, are shown as follows:

Number of ants  $m=8$ ; Weight of pheromone trail  $\alpha=1$ ; Weight of heuristic information  $\beta=2$ ; Pheromone evaporation parameter  $\rho=0.1$ ; Constant for pheromone updating  $Q=120$ ; the value of prior knowledge  $q_0=0.3$ ; the lower bound value of random search  $q_1=0.8$ ; Generation of evolution  $G=100$ .

TABLE I. PROCESS DATA OF THE ILLUSTRATION PROBLEM

Job	Operation	Machine					
		$M_1$	$M_2$	$M_3$	$M_4$	$M_5$	$M_6$
1	1	2	3	4	-	-	-
	2	-	3	-	2	4	-
	3	1	4	5	-	-	-
2	1	3	-	5	-	2	-
	2	4	3	-	-	6	-
	3	-	-	4	-	7	11
3	1	5	6	-	-	-	-
	2	-	4	-	3	5	-
	3	-	-	13	-	9	12
4	1	9	-	7	9	-	-
	2	-	6	-	4	-	5
	3	1	-	3	-	-	3

The Gantt chart for the best solution obtained by our improved ACO is illustrated, as shown in Fig.1. In addition, Fig.2 depicts the convergence curves of the best makespan. To such an instance, from Fig.2, it can be seen that our improved ACO obtains the best makespan with one generation of evolution. Therefore, the improved ACO is very efficient for solving FJSP.

In order to prove the solving solution quality and solving efficiency of our improved ACO, Table II shows the experimental results of the proposed improved ACO in comparison to other methods mentioned above by 20 independent experiments. The best solution, the average

convergence generation, the average convergence rate, the worst solution and the average computing time of our proposed algorithm are the best among the compared methods. The results demonstrate that the proposed ACO is effective and efficient in solving FJSP.

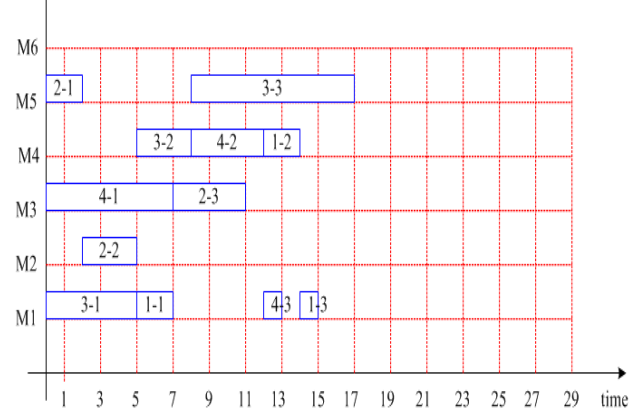


Figure 1. The best solution obtained by improve ACO algorithm (makespan=17).

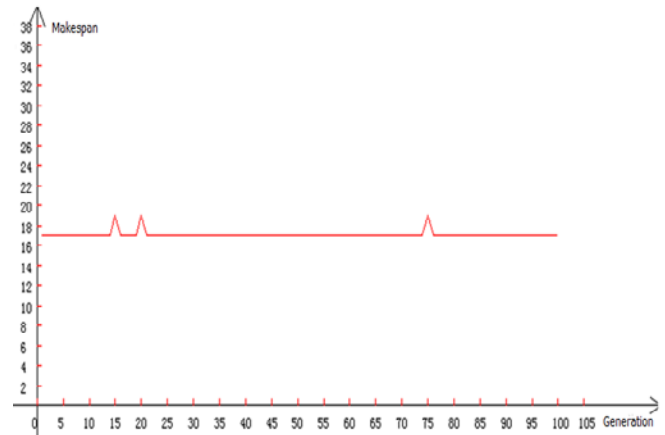


Figure 2. Convergence curve of the best solution.

TABLE II. EXPERIMENTAL RESULTS

Algorithm	The best solution	The average convergence generation	The worst solution	The average convergence rate (%)	The average computing time (s)
Ref.[15]	17	26.7	17.2	99.2	24.4
Ref. [16]	17	67.3	18	50	—
Ref. [17]	17	1	17	100	0.23
Our method	17	1.1	17	100	0.2192

## V. CONCLUSION

This paper investigates the FJSP with the objective of minimizing makespan. In the basic ACO, ant selected the next available machine by employing the transition probability does not always obtain the optimal direction and this would easily lead to the local optimal solution. Therefore a new node selection method which combining prior knowledge, probability search and random search is proposed. By this way, the search space can be increased and

the global optimal solution can be obtained with a larger probability. The experimental results show that the proposed improved ACO is able to find optimal or near-optimal solutions.

## ACKNOWLEDGEMENT

This paper is supported by the National Natural Science Foundation of China (Grant No. 51305001), Anhui Provincial Natural Science Foundation (Grant No. 1708085ME129), Key Support Projects for Outstanding

Young Talents of Anhui Province Universities (Grant No. gxyqZD2016125), and Anhui Key Laboratory Open Project of Advanced Numerical Control and Servo Technology (Grant No. xjsk003).

#### REFERENCES

- [1] M. Ziaee, "A heuristic algorithm for solving flexible job shop scheduling problem," *The International Journal of Advanced Manufacturing Technology*, Vol.71, No.1, 2014, pp.519-528.
- [2] A. Bagheri, M. Zandieh, I. Mahdavi, and M. Yazdani, "An artificial immune algorithm for the flexible job-shop scheduling problem," *Future Generation Computer Systems*, Vol.26, No.4, 2010, pp.533-541.
- [3] W. Bozejko, M. Uchroński, and M. Wodecki, "Parallel hybrid metaheuristics for the flexible job shop problem," *Computers & Industrial Engineering*, Vol.59, No.2, 2010, pp.323-333.
- [4] J. Q. Li, Q. K. Pan, P. N. Suganthan, and T. J. Chua, "A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem," *The International Journal of Advanced Manufacturing Technology*, Vol.52, No.5-8, 2010, pp.683-697.
- [5] T. Arit and P. Ajchara, "A hybrid artificial bee colony algorithm with local search for flexible job-shop scheduling problem," *Procedia Computer Science*, Vol.20, 2013, pp.96-101.
- [6] P. Brandimarte, "Routing and scheduling in a flexible job shop by tabu search," *Annals of Operation Research*, Vol.41, No.3, 1993, pp.157-83.
- [7] I. Kacem, S. Hammadi, and P. Borne, "Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic," *Mathematics & Computers in Simulation*, Vol.60, No.3-5, 2002, pp.245-276.
- [8] G. H. Zhang, X. Y. Shao, P. G. Li, and L. Gao, "An effective hybrid swarm optimization algorithm for multi-objective flexible job shop scheduling problem," *Computers & Industrial Engineering*, Vol.56, No.4, 2009, pp.1309-1318.
- [9] N. B. Ho and J. C. Tay, "Solving multiple-objective flexible job shop problems by evolution and local search," *IEEE Transactions on Systems Man and Cybernetics, Part C*, Vol.38, No.5, 2008, pp.674-685.
- [10] L. N. Xing, Y. W. Chen, and K.W. Yang, "An efficient search method for multi-objective flexible job shop scheduling problems," *Journal of Intelligent Manufacturing*, Vol.20, No.3, 2009, pp.283-293.
- [11] S. J. Wang and J. B. Yu, "An effective heuristic for flexible job shop scheduling problem with maintenance activities," *Computers & Industrial Engineering*, Vol.59, No.3, 2010, pp.436-447.
- [12] X. Y. Li, & L. Gao, "An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem," *International Journal of Production Economics*, 174, 2016, pp.93-110.
- [13] V. Roshanaci, A. Azab, and H. Elmaraghy, "Mathematical modeling and a metaheuristic for flexible job shop scheduling," *International Journal of Production Research*, Vol.51, No.20, 2013, pp.6247-6274.
- [14] P. Surekha and S. Sumathi, "Solving fuzzy based job shop scheduling problems using GA and ACO," *Journal of Emerging Trends in Computing and Information Sciences*, Vol.1, No.2, 2010, pp.95-102.
- [15] W. C. Zhang, P. Zhang, and X. D. Wu, "Solving flexible job-shop scheduling problems based on master-slave genetic algorithm," *Computer Integrated Manufacturing System*, Vol.12, No.8, 2006, pp.1241-1245.
- [16] X. M. Yang and J. C. Zeng, "Solving flexible job shop scheduling problem using genetic algorithm," *Control and Decision*, Vol.19, No.10, 2004, pp.1197-1200.
- [17] G. H. Zhang, "Research on the method of flexible job shop scheduling," Wuhan: Huazhong University of Science and Technology, 2009.