

# An New Estimation of Distribution Algorithm Based Edge Histogram Model for Flexible Job-Shop Problem

Xiaojuan He<sup>1,2</sup>, Jianchao Zeng<sup>2</sup>, Songdong Xue<sup>2</sup>, Lifang Wang<sup>2</sup>

<sup>1</sup> College of Electrical and Information Engineering, Lanzhou University of Technology,  
Lanzhou, China

<sup>2</sup> Complex System and Computational Intelligence Laboratory,  
Taiyuan University of Science and Technology, Taiyuan, China  
hexjxian@gmail.com, zengjianchao@263.net,  
{xuesongdong,wlf1001}@163.com

**Abstract.** An estimation of distribution algorithm for flexible job shop scheduling problem was proposed. The probability model was given using frequency information of pair-wise operations neighboring. Then the structure of optimal individual was marked and the operations of optimal individual were partitioned to some independent sub-blocks. Each sub-block was taken as a whole to be adjusted to avoid repeating search in same area and improve search speed. The experimental results show that the proposed algorithm is efficient for Flexible Job-Shop Problems.

**Keywords:** FJSP, Estimation of Distribution Algorithms, Neighboring Operations, Probability Model.

## 1 Introduction

The job-shop scheduling problem is one of the most popular manufacturing optimization models used in practice. Flexible job shop problem (FJSP) is a generalization of the job shop that allows one operation to be processed on any machine from a set of alternative machines. It is closer to the real manufacturing situation. The extension includes two tasks: Assignment operations to appropriate machines and sequencing the operations on each machine. These features of the FJSP obviously increase the complexity of finding optimal solutions even of approximately optimal solutions. So the task is more challenging than the classical job shop[1,2]. In recent years, meta-heuristic techniques such as simulated annealing, tabu search and evolutionary algorithms have been adopted to solve the FJSP to find a promise solution and many promising results have been obtained [3, 4, 5].

A new evolutionary algorithm based on probability analysis, i.e., estimation of distribution algorithms (EDAS) has become very popular. The algorithm relies on the construction and maintenance of a probability model that characterizes satisfactory solutions for a problem. And this probabilistic model is used to guide further exploration of search space and realize evolution process. EDAS effectively overcomes the problem of blocks being disrupted, has shown to perform very well on a wide variety

of problem that traditional GA difficult to solved, especially on high dimensional complex problems [6, 7].

In this paper, we propose an efficient method to solve the flexible job shop problems. The probability model that can well reflect the character of the flexible job shop scheduling problem is built and improves the ability of edas to solve the flexible job shop scheduling problem. This paper is organized as follows: In Section 2, we shortly describe the basic framework of EDAS and FJSP. Section 3 gives a detail description of new proposed algorithm for FJSP. Section 4 discusses the experimental results. Finally, Section 5 summarizes the contribution of this paper.

## 2 Problems Description

### 2.1 The General Framework of EDAS

EDAS is a class of novel evolutionary algorithms. The algorithms represent relations between the variables by building probability model [6]. Main steps are summarized as follows:

Step 1 Generate initial population.

Step 2 Evaluate population. Select some better individuals as superior population. Construct the probability model according to the information of superior population.

Step 3 Sampling from the probability model constructed and generates new population, evaluate new population. Select new superior population.

Step 4 Update the probability model.

Repeatedly do the step 3 and step 4 until the termination criteria is met.

### 2.2 FJSP Formulation

The FJSP may be stated as follows [1, 2]: there are  $n$  jobs to be processed on machines. Each job consists of a predetermined sequence of operations. Each machine can process only one operation at a time interval. Each operation must be completed without interruption once started. The objection of FJSP is to assign each operation to an appropriate machine and to sequence the operations on corresponding machines in order to minimize the make span.

## 3 EDAS for FJSP

### 3.1 Encoding and Decoding Methods

Machine encoding way adopts an array of the number of alternative machines. We select a machine from the alternative machine set of each operation and all machines selected consist of a machine assignment individual. For example, in a FJSP, there is 4 jobs, and each job consists of 3 operations, each operation of each job has 4 alternative machines. So an individual is denoted as 12 numbers. The machine assignment individual is obtained as Fig.1:

We use an operation-based encoding method as operation encoding. All operations for a job are named with the same symbol and then interpret their meaning according to the order of appearance in sequence of chromosome. A operation individual (1 3 1

4 2 3 4 3 1 2 4 2) is generated randomly, each job will appear 3 times exactly in a operation individual. The corresponding relationships are as Fig.2. In principle, a chromosome can be decoded into a lot of schedules for the FJSP problem. In this paper we use priority-based decoding. The priority-based decoding method allows an operation to search the earliest available time interval between the operations already scheduled on its assigned machine. Some operations can be started earlier in time without altering the operation sequence and without delaying any other operations.

Operations	Indicated	$O_{11}$	$O_{12}$	$O_{13}$	$O_{21}$	$O_{22}$	$O_{23}$	$O_{31}$	$O_{32}$	$O_{33}$	$O_{41}$	$O_{42}$	$O_{43}$
		↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Machine	Assignment	1	3	4	3	2	2	1	3	4	4	3	2

**Fig. 1.** Illustration of the machine assignment

Operation Number	Indicated:	1	3	1	4	2	3	4	3	1	2	4	2
		↓	...	↓			...			↓		...	
Operation Sequence :		$O_{11}$	$O_{31}$	$O_{12}$	$O_{41}$	$O_{21}$	$O_{32}$	$O_{42}$	$O_{33}$	$O_{13}$	$O_{22}$	$O_{43}$	$O_{23}$

**Fig. 2.** Illustration of the operation sequence

### 3.2 Population Initialization and Fitness Evaluate

Firstly, the machine assignment individuals are generated. In order to enhance the assignment performance, machine is assigned for each operation which processes this operation with a minimal processing time or the near-minimal processing time. Secondly, generate initialization operation population. Let  $\pi = (\pi(1), \pi(2), \dots, \pi(n \times m))$  be an integer permutation form 1 to  $n \times m$ . Let the population size be  $N$ .  $N$  integer arrays are randomly generated and consist of the initial operation population. Evaluate the fitness of individuals. Some better individuals are selected as superior population according to some proportional. Let the superior population size be  $D$ .

### 3.3 Construct and Update Probability Model

In this paper, we have constructed probability model according to the order of operations of individuals in superior population based on edge histogram [7]. Let the initial superior population be represented as  $D^0$ . The  $k$ th individual is as  $\chi_k^0 = (\pi_k^0(1), \pi_k^0(2), \dots, \pi_k^0(n \times m))$ , set  $k \in D$ . Through calculating the frequencies of pair-wise neighboring operations appearing in superior population  $D^0$ , the probability matrix is constructed as follows:

$$P^t = \begin{bmatrix} p_{11}^t & p_{12}^t & \cdots & p_{1,n \times m}^t \\ p_{21}^t & p_{22}^t & \cdots & p_{2,n \times m}^t \\ \cdots & \cdots & \cdots & \cdots \\ p_{n \times m, 1}^t & p_{n \times m, 2}^t & \cdots & p_{n \times m, n \times m}^t \end{bmatrix}$$

Where  $t$  represents the iterative number, and when  $t=1$ , let

$$p_{ij}^t = (1 - \alpha) \cdot \frac{1}{n \times m} + \frac{1}{D} \sum_{k=1}^D \delta_{ij}(x_k^0) \quad i, j = 1, 2, \dots, n \times m \quad (1)$$

$$\delta_{ij}(x_k^0) = \begin{cases} 1 & \pi_k^0(h) \rightarrow o_i \wedge \pi_k^0(h+1) \rightarrow o_j \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Where  $h \in \{1, 2, \dots, (n \times m - 1)\}$ ,  $\delta_{ij}(x_k^0)$  equals to 1 denotes that the neighboring operation  $o_i, o_j$  appear in the structure of the  $k$ th individual. Otherwise  $\delta_{ij}(x_k^0)$  equals to 0. The parameter  $\alpha$  is a learning rate, and  $\alpha \in (0, 1)$ .

At iteration  $t$  ( $t > 1$ ), the new probability matrix is built through  $D^t$ . The probability is updated as follows:

$$p_{ij}^{t+1} = (1 - \alpha) \cdot p_{ij}^t + \alpha \cdot \frac{1}{D} \sum_{k=1}^D \delta_{ij}(x_k^t) \quad i, j = 1, 2, \dots, n \times m \quad (3)$$

### 3.4 The Process of Block Optimization

If there are multi-neighboring operations, their frequency appearing in superior population is higher. Then in order to avoid the multi-neighboring operations being destroyed and repeating search in same space in iterative process, these operations are connected into a whole block and take part in iterative process. At iteration  $l$ , according to corresponding  $p^l$ , the model of block structure is built by dividing to the structure of optimal individual. Here, a parameter  $\partial$  is set advanced.  $\partial$  means a connection condition, set  $\partial \in (0, 1)$ . If the frequency of multi-neighboring operations appearing in superior population is bigger than the parameter  $\partial$ , then these neighboring operations can be connected into a block. The main process is as follows: denotes the optimal individual as  $\chi^* = (\pi_*^l(1), \pi_*^l(2), \dots, \pi_*^l(n \times m))$ , and set  $\pi_*^l(h) \rightarrow o_i \wedge \pi_*^l(h+1) \rightarrow o_j$ . Corresponding to the matrix  $p^l$ , we consider that the operation  $o_i$  and  $o_j$  have relativity and belong to a same sub-block if  $p_{ij}$  is higher than  $\partial$ . Otherwise they belong to different sub-block. Then they being separately as a whole take part in the next iterative process. Generally empirical set  $\partial \in (0.7, 0.9)$ . The sequence among different sub-blocks is adjusted according some probability.

### 3.5 Generate New Population

The new population is generated by roulette wheel sampling from probability matrix  $P^l$ . The sampling method is as follows: Randomly select a initial point (operation),  $o_i$  for example, the next operation is selected from the  $i$ th row of probability matrix  $P^l$  by roulette wheel. If the selected next operation is  $o_j$ , then the next operation is selected again from the  $j$ th row of probability matrix  $P^l$  using roulette wheel, etc. Then go to 3.3.

Do repeatedly the above step from 3.3 to 3.5 till the termination criterion is met. The elitist strategy is applied to the whole iterative process for ensuring the convergence of algorithm.

4 Computational Results

*Problem 4×5* To illustrate the effectiveness and performance of our proposed algorithm in this paper, a small scale instance is selected to compute [1, 2]. The population size is 30 and the most iteration number is 50.  $W_k$  denotes the workload of the most loaded machine.  $W_t$  denotes total workload of machines. Let the proportion of the superior population selected be 0.3, and set  $\alpha=0.15$ ,  $\vartheta=0.85$ . When the fitness value of the best solution is not improved after the continual 5 times in iteration process, the process of block optimal is executed. After the process of block optimal iterates 5 times, return the next iterative process. Each instance is executed for 10 runs. The obtained optimal solution is summarized as Fig.3 and Table 1.

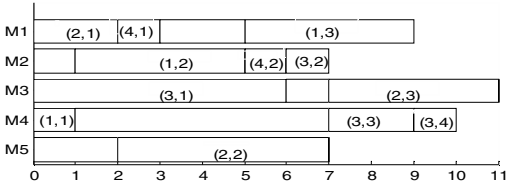


Fig. 3. The Gantt chart of the obtained optimal solution for problem 4×5

Table 1. Comparison of results on problem 4×5

	AL+CGA	PSO+TS	Edas
Makespan	16	11	11
Wk	10	10	10
Wt	34	32	32

From the result of Table 1, we can see that the obtained results of EDAS proposed in this paper are better than hybrid algorithm AL+CGA [1, 2], and are similar with PSO+TS [5]. The computation results show that the EDAS proposed in this paper has better search ability.

*Other Instances.* The algorithm is tested on three large scale instances from literature [1]. The problem 8×8 is an instance of partial flexibility. We give the processing time on impossible machine a big number such as 9999 then make the partial flexibility problem be converted to a total flexibility problem. The population size is 50 and the most iteration number is 100. The other parameters are same as the above. The result is compared with other famous algorithms on makespan as Table 2.

From the result of Table 2, we can see that the obtained results of EDAS proposed in this paper are better than hybrid algorithm AL+CGA, PSO+SA, moga and hga [3],

**Table 2.** Comparison of results between different algorithms

Problem size	Algorithm						
	AL+CGA	PSO+SA	Moga	Hga	PVNS	PSO+TS	Edas
8×8	15	15	15	15	14	14	14
10×10	7	7	7	7	7	7	7
15×10	24	12	12	12	12	11	12

at the same time, The obtained results of EDAS proposed in this paper are similar with algorithm PVNS [3] and PSO+TS. It shows that the EDAS which don't hybrid other local search algorithm in this paper has well search ability.

5 Conclusions

In this paper, we adopt EDAS to solve the flexible job shop scheduling problem. The probability model is built through considering the information of neighboring operations appearing in superior population. This makes the probability model well reflect the character of flexible job shop scheduling problem, and improves the ability of edas to solve flexible job shop scheduling problem. The simulation results show that the proposed algorithm is efficient.

**Acknowledgments.** This paper is supported by The Shanxi Science Foundation for Young Scientists under Grant 2010021017-2, China.

References

1. Kacem, I., Hammadi, S., Borne, P.: Approach by Localization and Multiobjective Evolutionary Optimization for Flexible Job-Shop Scheduling Problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 32, 1–13 (2002a)

2. Kacem, I., Hammadi, S., Borne, P.: Pareto-Optimality Approach for Flexible Job-Shop Scheduling Problems: Hybridization of Evolutionary Algorithms and Fuzzy Logic. *Mathematics and Computers in Simulation* 60, 245–276 (2002b)

3. Yazdani, M., Amiri, M., Zandieh, M.: Flexible Job-Shop Scheduling with Parallel Variable Neighborhood Search Algorithm. *Expert Systems with Applications* 37, 678–687 (2010)

4. Gao, J., Gen, M., Sun, L.Y., Zhao, X.H.: A Hybrid Of Genetic Algorithm and Bottleneck Shifting for Multiobjective Flexible Job Shop Scheduling Problems. *Computers & Industrial Engineering* 53, 149–162 (2007)

5. Zhang, G.H., Shao, X.Y., Li, P.G., Gao, L.: An Effective Hybrid Particle Swarm Optimization for Multi-Objective Flexible Job-Shop Scheduling Problem. *Computers & Industrial Engineering* 56, 1309–1318 (2009)

6. Pelikan, M., Goldberg, D.E., Lobo, F.: A Survey of Optimization by Building and Using Probabilistic Models. *Illigal Report, Urbana, illinois* (1999)

7. Tsutsui, S.: Probabilistic model-building genetic algorithms in permutation representation domain using edge histogram. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañás, J.-L., Schwefel, H.-P. (eds.) *PPSN 2002. LNCS*, vol. 2439, pp. 224–233. Springer, Heidelberg (2002)