

Genetic Algorithm Combined with Gradient Information for Flexible Job-shop Scheduling Problem with Different Varieties and Small Batches

Ming Chen¹, Jie-Lin Li²

¹Sino-German College of Applied Sciences, Tongji University, Shanghai 201804, China

²School of Mechanical Engineering, Tongji University, Shanghai 201804, China

Abstract. To solve the Flexible Job-shop Scheduling Problem (FJSP) with different varieties and small batches, a modified meta-heuristic algorithm based on Genetic Algorithm (GA) is proposed in which gene encoding is divided into process encoding and machine encoding, and according to the encoding mode, the machine gene fragment is connected with the process gene fragment and can be changed with the alteration of process genes. In order to get the global optimal solutions, the crossover and mutation operation of the process gene fragment and machine gene fragment are carried out respectively. In the initialization operation, the machines with shorter manufacturing time are more likely to be chosen to accelerate the convergence speed and then the tournament selection strategy is applied due to the minimum optimization objective. Meanwhile, a judgment condition of the crossover point quantity is introduced to speed up the population evolution and as an important interaction bridge between the current machine and alternative machines in the incidence matrix, a novel mutation operation of machine genes is proposed to achieve the replacement of manufacturing machines. The benchmark test shows the correctness of proposed algorithm and the case simulation proves the proposed algorithm has better performance compared with existing algorithms.

1 Introduction

With the transformation from manufacturing industry into service industry, the production process under the intelligent manufacturing environment should satisfy the customization and individuation demands of the customer, which has brought the requirement of scheduling problem with different varieties and small batches [1-2]. Compared with traditional Job-shop Scheduling Problem (JSP), it is much harder to achieve the global optimal solutions of FJSP because there exist some redundancies of the manufacturing resources, which causes the rapid increase of search space with the expansion of the problem scale [3-6].

As a kind of Non-deterministic Polynomial-hard (NP-hard) problems, there are mainly two directions of solution methods: exact methods and approximate methods [7]. The exact methods such as mathematical programming [8], branch and bound algorithm [9-10] can obtain the final theoretical optimal solutions of FJSP with small scale. For the large-scale FJSP, this kind of exact methods is not suitable because the expansion of problem scale brings exponential explosion, which takes much more time cost using such enumeration algorithms [11]. To solve this kind of large-scale scheduling problems with lower time cost, corresponding to the exact methods, the approximate algorithms such as GA [12-14], Particle Swarm Optimization (PSO) [15-18], Binary Differential Evolution Harmony Search [19] and Simulated Anneal algorithm [20] have been integrated. These meta-heuristic algorithms maybe cannot get the theoretical global

optimal solutions of large-scale FJSP, but the obtained solutions can meet the actual production requirement. In these meta-heuristic algorithms, GA is adapted to solve such highly discrete problem like FJSP very well since the encoding and decoding mode of GA can reflect the discrete solutions. But in the evolutionary process of GA, the new-generated populations may not satisfy the constraint requirement owing to the gradient information loss between the father population and child population. To improve the controllability of evolutionary process and ensure the convergence speed, a GA-based algorithm is proposed with high solving efficiency and optimization performance.

2 Mathematical model

FJSP with different varieties and small batches can be described as follows:

In order to produce N kinds of products with job sum J , the i th workpiece of k th assortment should be processed according to the process order O_{kj} on the selected machine, in which $1 \leq i \leq J$, and $1 \leq j \leq n_i$, n_i means the process amount of the i th workpiece. Meanwhile, $J = \sum_{k=1}^N P_k$, P_k denotes the batch amount of the k th assortment and the operation time T_{ijl} on a concrete machine l is already defined. So the incidence

matrix J_o between J and process order O as well as the time matrix T can be obtained. Based on these two matrixes, the optimization model can be established.

The whole production process with different varieties and small batches ends once the last process on a concrete machine is finished, so the minimum value of max make span is utilized as the optimization objective to dispatch the manufacturing resources. So the fitness function in optimization model can be summarized in following Eq. (1):

$$f(x) = \max(T_m) \quad (1)$$

in which $1 \leq m \leq M$, M is the sum number of alternative machines.

Problem constraints are as follows:

- The next process of any job can start only the current process is finished;
- One machine can only process one workpiece at the same time;
- Once the process starts it cannot be suspended until it is finished;
- One workpiece can only be processed on one machine at the same time;
- The workpieces of same kind have the same process order;
- The situation of machine fault is out of consideration.

3 GA-based algorithm design

Fig. 1 depicts the computation framework of proposed algorithm based on GA. Different from traditional GA, the encoding genes are divided into process gene fragment and machine gene fragment which have close relationship with each other, and the following evolutionary operations are all based on these two kinds of genes. The specific operations are described in detail in the following sections.

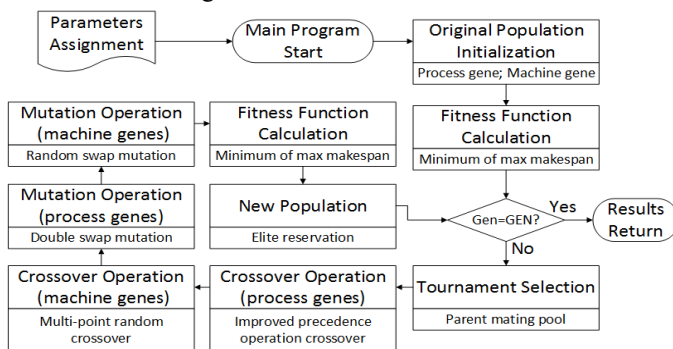


Figure 1. Tournament selection and elite reservation

3.1 Population initialization

Aiming at solving double decision-making problems of the job processes and machines under production demand of different varieties and small batches, the chromosome is composed of two kinds of gene fragments: the process genes and machine genes. The first process gene fragment means the process order on the alternative machines according to the manufacturing procedures of the given job. The latter machine gene fragment means

the machine number in the alternative machine sets of the corresponding process order. In order to maintain the consistency with the solutions, decimal coding is applied to compose the population chromosome.

(1) For the process genes, a gene fragment composed of J genes is initialized, and the value of each gene means the process order of the i th work piece.

(2) For the machine gene, an initialization strategy is applied to accelerate the algorithm convergence speed because the machines with shorter manufacturing time are more likely to be chosen. The max time difference of every non-empty positions in the time matrix T is calculated and sorted in ascending order. The machine genes of the first part are initialized randomly within the range of corresponding positions in the incidence matrix J_o , and for the latter part, the machines with minimum manufacturing time are chosen as the machine genes.

3.2 Tournament selection and elite reservation

The classic roulette selection is not suitable because the optimization objective is to minimize the max make span of a given job and some information of chromosome may

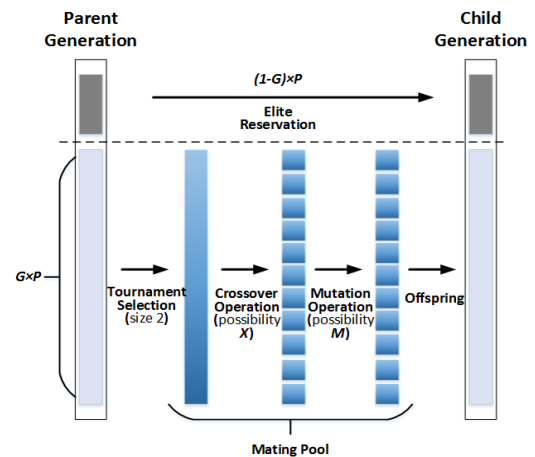


Figure 2. Tournament selection and elite reservation

lose in the transmission operation. Therefore, tournament selection strategy is utilized to select the superior offspring. At a time two individuals are chosen randomly from the parent generation, and the individual with a smaller fitness function is put into the mating pool.

In order to ensure the superior offspring can continue the genetic procedure instead of elimination, an elite reservation strategy using generation gap mechanism is proposed. For every evolutionary iteration, the mating pool consists of parent chromosomes selected by tournament strategy with quantity $G \times P$, in which G means the generation gap coefficient, and P means the population quantity. Fig. 2 shows the important role of tournament selection and elite reservation in the pretreatment phase of the evolutionary process. From Fig. 2, the parent chromosomes with the proportion of $(1-G)$ are selected to be the offspring directly.

3.3 Crossover operation

3.3.1 Crossover operation of process genes

The Improved Precedence Operation Crossover (IPOX) developed by Zhang CY [21] is applied to achieve the crossover operation of process genes. More detail information can be found in reference [21] and will not be described here. From the population initialization process, we can see the process genes and machine genes have a very close connection between each other and the loci of the machine genes should follow the change of the process genes, so after the crossover operation of the process genes, the corresponding machine genes must execute the gene loci based on the change of the process genes.

3.3.2 Crossover operation of machine genes

Different from single-point crossover, a multi-point crossover strategy is introduced to achieve the crossover operation of machine genes and to accelerate the evolutionary speed, the quantity of crossover points is related with the difference degree between machine gene fragments in different parent individuals directly. When the machine genes at the same gene locus of process gene fragment are different, the corresponding genes should be interchanged if the judgment condition $rand(0,1) > 0.5$ is satisfied.

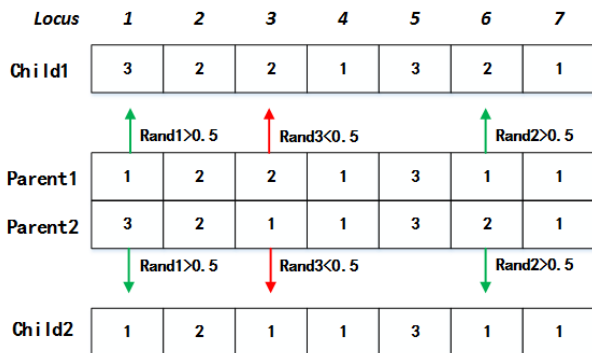


Figure 3. Multi-point random crossover operation.

Fig. 3 shows a simple crossover execution example of a concrete machine gene fragment. There are three gene loci with different genes: locus 1, 3 and 6. The genes at loci 1 and 6 are interchanged with each other between the two parent chromosomes to achieve the crossover operation because the random condition is qualified but the genes at locus 3 of both parent chromosomes should remain without any change.

3.4 Mutation operation

3.4.1 Mutation operation of process genes

The genetic procedure depends much on the gradient evolutionary information because the arbitrary change between two elements may generate invalid child individuals which cannot satisfy the constraint conditions. Therefore, the double mutation operation should be limited under a certain condition. Supposing double

selected gene loci are $L1$ and $L2$, the genes of these double loci can be interchanged only if the genes are different and there is no same gene as that of $L1$ or $L2$ in the gene fragment between these two gene loci. This limitation can avoid destroying the process order of any workpiece so that the validity of child populations is guaranteed.

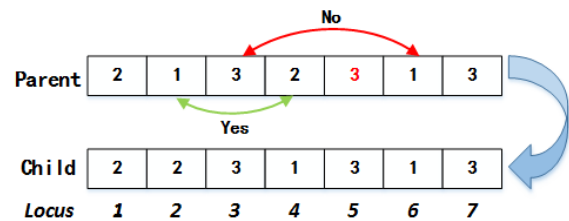


Figure 4. Double swap mutation operation.

Fig. 4 shows a simple mutation example of the process genes. The genes at loci 3 and 6 cannot be interchanged because there exists a same process number 3 between these two loci, and the genes at loci 2 and 4 can carry out the mutation operation because the mutation condition is satisfied.

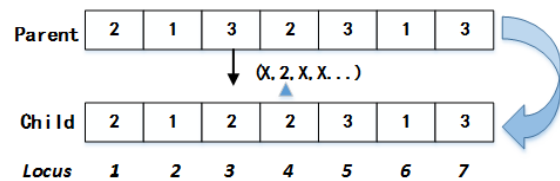


Figure 5. Random swap mutation operation

3.4.2 Mutation operation of machine genes

The mutation operation of machine genes is very important because this operation builds up the interchange relationship between the current machine with the alternative machines prepared in the incidence matrix J_o . For a randomly selected machine gene in the gene fragment, a random machine in the alternative machine set of corresponding position in J_o is picked out to replace the current machine gene.

Fig. 5 depicts a mutation example of machine genes. The gene at locus 3 is randomly selected to carry out the mutation operation and then the machine with number 2 is picked out from the alternative machine set in the incidence matrix J_o . The child chromosome is generated as shown from the parent generation finally.

3.5 Self-adaptive strategy

Compared with the crossover operation of the gene fragments between the parent chromosomes, the mutation operations of process genes and machine genes are miniature, so a self-adaptive mechanism is introduced to adjust the mutation times. Hereon, self-adaptive operator $\text{floor}(\log_{10}(G_{EN} - G_{en})) + 1$ is utilized to control the mutation procedure, in which floor means the floor

function, G_{en} means the current population generation and G_{EN} means the termination criteria. At the beginning of the evolutionary process, this parameter can control the mutation times in the every generation process of offspring, which can improve the chances of better offspring and accelerate the evolutionary rate. With the evolutionary process, this parameter helps the population converge to the optimal results.

4 Experiments and discussion

The simulation experiments are carried out on Windows 7 platform with the hardware parameters as follows: CPU Intel E8400 3.00GHz, RAM 2GB. The programming environment is MatLab R2013a. According to the implement procedure in Fig. 1, the parameters are assigned as follows: population size $P = 100$, termination criteria $G_{EN} = 200$, generation coefficient $G = 0.9$, crossover possibility $X = 0.8$ and mutation possibility $M = 0.6$.

4.1 Benchmark test

The global optimal solutions of scheduling case with different batch scales are as shown in Table 1, and the detail parameters of benchmark test can be found in reference [22]. In this case, there are four kinds of product demands with different batch sizes, and the theoretical solutions as follows. The benchmark test shows that the proposed algorithm can also obtain the global results, which proves the correctness of proposed algorithm.

Table 1. Test results of proposed algorithm

Problem Size	P_1	P_2	P_3	P_4	Optimal Solutions	Obtained Solutions
1x4	1	1	1	1	17	17
2x4	2	2	2	2	34	34
3x4	3	3	3	3	51	51
5x4	5	5	5	5	85	85
10x4	10	10	10	10	170	170

Additionally, the optimized makespan of case 4 in reference [22] is 33, but the result obtained by proposed algorithm 32, which shows the priority of proposed algorithm applied in the scheduling optimization problems. The concrete scheduling scheme can be seen in Fig. 6.

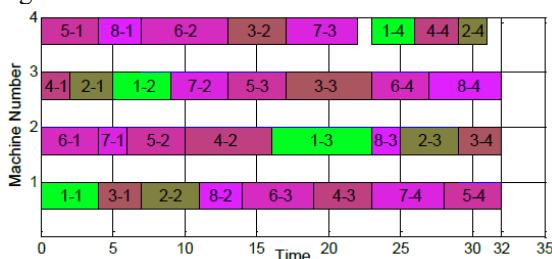


Figure 6. Gantt diagram of optimized scheduling result

4.2 Case study

For the case of FJSP with different varieties and small batches presented by Fu WP, the final optimized solution is 121 [23], but the solution got by the proposed algorithm is 113, which shows the superiority clearly. The convergence procedure is as shown in Fig. 7. It can be seen that the fitness function value converges from 181 to 113 eventually, which is a new lower bound compared with the result in reference [23]. Meanwhile, the proposed algorithm can obtain the solution which can satisfy the industrial scheduling requirements with an acceptable computation cost with 86 evolutionary times.

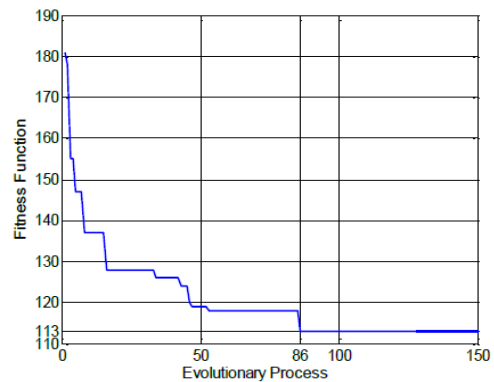


Figure 7. Fitness function with respect to evolutionary process.

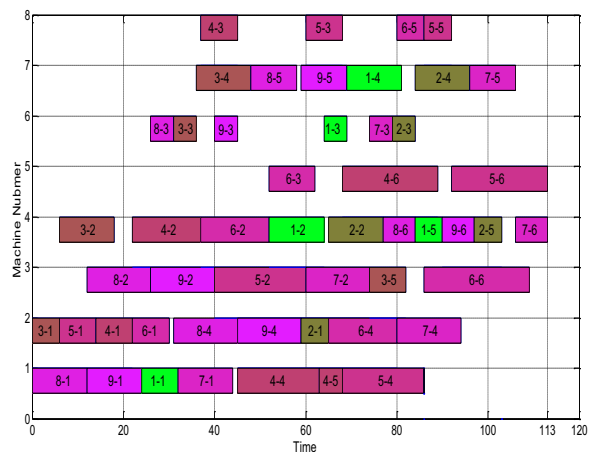


Figure 8. Gantt diagram of optimized scheduling result

The corresponding Gantt diagram in Fig. 8 shows the scheduling of manufacturing machines according to the job requirements with different varieties and small batches. The global optimal solution of this case has not been found until now, and the results with different scheduling schemes got by proposed algorithm are the minimum solutions for the moment. The next research step focuses on finding out the theoretical global solution or updating a new lower bound again.

5 Conclusions

A modified scheduling algorithm based on GA is proposed to solve the FJSP with different varieties and small batches. The initialization operation associated with time matrix is introduced to accelerate the convergence speed and the generation gap coefficient is applied to guarantee the survival rate of superior offspring. To

maintain the population diversity and avoid falling into local optimal solutions, IPOX and multi-point random crossover operation are utilized to achieve the crossover of chromosomes, and double swap mutation operation of the process genes and random swap mutation operation of the machine genes are introduced to realize the variation evolution. The benchmark test shows the correctness of proposed algorithm and the case simulation proves the proposed algorithm has better performance compared with existing algorithms. The benchmark tests and simulation results show the proposed algorithm can obtain the global solutions and update the new lower bound of FJSP with different varieties and small batches got by existing algorithms.

References

1. Lee J, Kao HA, Yang S (2014) Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia CIRP*, 16: 3-8
2. Brettel M, Friederichsen N, Keller M et al (2014) How virtualization, decentralization and network building change the manufacturing landscape: An Industry 4.0 Perspective. *International Journal of Mechanical, Industrial Science and Engineering*, 8(1): 37-44
3. Wang Y (2012) A new hybrid genetic algorithm for job shop scheduling problem. *Computers & Operations Research*, 39(10): 2291-2299
4. Nouri HE, Driss OB, Ghédira K (2015) Aholonicmultiagent model based on a combined genetic algorithm-tabu search for the flexible job shop scheduling problem. In *Highlights of Practical Applications of Agents, Multi-Agent Systems, and Sustainability-The PAAMS Collection*, pp 43-54
5. Wang L, Wang S, Xu Y et al (2012) A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem. *Computers & Industrial Engineering*, 62(4): 917-926
6. Wang J, Chu K (2012) An application of genetic algorithms for the flexible job-shop scheduling problem. *International Journal of Advancements in Computing Technology*, 4(3)
7. Li J, Pan Q, Xie S (2012) An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems. *Applied Mathematics and Computation*, 218(18): 9353-9371
8. Baptiste P, Le PC, Nuijten W (2012) *Constraint-based scheduling: applying constraint programming to scheduling problems (Vol. 39)*. Springer Science & Business Media
9. Brucker P, Burke EK, Groenemeyer S (2012) A branch and bound algorithm for the cyclic job-shop problem with transportation. *Computers & Operations Research*, 39(12): 3200-3214
10. Liaw, CF (2013) An improved branch-and-bound algorithm for the preemptive open shop total completion time scheduling problem. *Journal of Industrial and Production Engineering*, 30(5): 327-335
11. Fribourg L, Soulat R, Lesens D et al (2012) Robustness analysis for scheduling problems using the inverse method. In *Temporal Representation and Reasoning (TIME)*, 2012 19th International Symposium on pp 73-80
12. Chen JC, Wu CC, Chen CW et al (2012) Flexible job shop scheduling with parallel machines using Genetic Algorithm and Grouping Genetic Algorithm. *Expert Systems with Applications*, 39(11): 10016-10021
13. Chirici L, Wang KS (2014) Tackling the storage problem through genetic algorithms. *Advances in Manufacturing*, 2(3): 203-211
14. Nouri HE, Belkahla DO, Ghedira K (2015) Genetic algorithm combined with tabu search in a holonicmultiagent model for flexible job shop scheduling problem. In *17th International Conference on Enterprise Information Systems (ICEIS)*
15. Tavakkoli-Moghaddam R, Azarkish M, Sadeghnejad-Barkousaraie A (2011) A new hybrid multi-objective Pareto archive PSO algorithm for a bi-objective job shop scheduling problem. *Expert Systems with Applications*, 38(9): 10812-10821
16. [16] Yu Q, Wang K (2014) A hybrid point cloud alignment method combining particle swarm optimization and iterative closest point method. *Advances in Manufacturing*, 2(1): 32-38
17. Moslehi G, Mahnam M (2011) A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. *International Journal of Production Economics*, 129(1): 14-22
18. Wang KS (2013) Towards zero-defect manufacturing (ZDM)—a data mining approach. *Advances in Manufacturing*, 1(1): 62-74
19. Wang L, An L, Ni HQ et al (2016) Pareto-based multi-objective node placement of industrial wireless sensor networks using binary differential evolution harmony search. *Advances in Manufacturing*, 4(1): 66-78
20. Elmi A, Solimanpur M, Topaloglu S et al (2011) A simulated annealing algorithm for the job shop cell scheduling problem with intercellular moves and reentrant parts. *Computers & industrial engineering*, 61(1): 171-178
21. Zhang CY, Dong X, Wang XJ et al (2010) Improved NSGA-II for the multi-objective flexible job-shop scheduling problem. *Journal of Mechanical Engineering*, 46(11): 156-164
22. Zhang XX, Zhang WL (2010) FJSP based on timed Petri Net and GA. *Systems Engineering*, (11): 86-94
23. Fu WP, Liu DM, Lai CW et al. (2011) Polychromatic-sets-based improved genetic algorithm for solving multi-species FJSP. *Computer Integrated Manufacturing Systems*, 17(5): 1004-1010