Optimization Model for Job Shop Scheduling Based on Genetic Algorithm

Yu-jie Zhu and Ya-min Liang

Abstract JSSP is short for Job-shop Scheduling Problem. With multi-constraint, multi-objective and random uncertainty, JSSP is optimization problems and also the most difficult constraints combinatorial optimization problems. JSSP is an important aspect of modern enterprise manufacturing operations under advanced manufacturing mode. As an effective mean of solving complex scheduling problems in many areas related to scheduling, Genetic Algorithm (GA) has been effectively applied. With random, highly parallel and adaptive intelligent features, GA has showed its unique advantages in solving problems with complex combination of multiple constraints, such as JSSP. This article will establish an optimization model for JSSP based on genetic algorithm and solve the model with MATLAB, and then output the result of the scheduling with a Gantt chart. There will be an example to verify the feasibility and effectiveness of the algorithms and models.

Keywords GA • Gantt chart • JSSP • Optimization model

1 Introduction

JSSP which is short for Job-shop Scheduling Problem tend to be complex, dynamic random, multi-objective and multi-constrained, motion blur and other characteristics and so on (Junqiang Wang et al. 2005). It needs to give the following assumptions in order to describe JSSP more clearly:

1. The machining of the parts must meet their own specific requires of process route. There are strict order of the various processes and requirements of the specified

Y. Zhu • Y. Liang (✉)
Department of Industrial Engineering, Northeast Forestry University, Harbin,
People's Republic of China
e-mail: liangyamin0618@139.com

E. Qi et al. (eds.), *Proceedings of 20th International Conference on Industrial Engineering and Engineering Management*, DOI 10.1007/978-3-642-40063-6_85,

machine for the same parts. It can not cross a process and carry out the next step of processing;
2. At one point, for example, a machine and a part can only be a one-to-one correspondence. It does not appear the phenomenon that one machine are machining two parts at the same time;
3. The process route and process lead time of the parts are known and fixed;
4. It does not appear interrupts and machine failure in the whole processing;
5. Each part can only go through a given machine only one time;
6. All machines are different together with the type of the processing (Fuxiang Zhang et al. 2009).

Genetic algorithm (GA) has been effectively applied in many areas related to scheduling as an effective mean of solving complex scheduling problems (Guohui Zhang et al. 2009). Typically, the search process of GA begins at a set of randomly generated initial "population". Each individual in the population of every generation represents a solution to the problem. This individual is called "chromosome" as a bridge to connect genetic algorithms and problem solution. Each chromosome is actually an individual encoded by the gene. Chromosome produces new individuals through "genetic manipulation". All new individuals make up a new population. In the chromosome population of each generation, there is a standard to measure the quality of chromosomes. That is "fitness value" which can measure the degree of chromosome adapt to the environment. It is used as the basis for the individual selecting. Individuals can continue to survive after going through the selection operation and then generate new populations through the "combination", "cross" and "alternative" to establish a new solution set. As the same with the epigenetic generation by natural selection, new populations that go through this process will adapt to the environment more than the previous generation. We can get the approximate optimal solution to the problem through "decoding" the best individual in the last generation of the population (Tienan Zhang et al. 2011).

2 Operational Processes of GA

Typically, GA contains some typical stages such as production of the initial population, crossing, variation, selection. As shown in Fig. 1, the main steps (Ying Su et al. 2010):

(1) Encoding pairs of chromosomes according to the required solution of the problem,
(2) Generate a set of initial population randomly;
(3) Calculate fitness value of individuals in the initial population;
(4) Judge whether the termination condition is satisfied (convergence criterion), if the termination condition is satisfied, stop the operation, output the optimal solutions, otherwise, continue the next procedure;
(5) Carry out the selection procedure according to the fitness value;

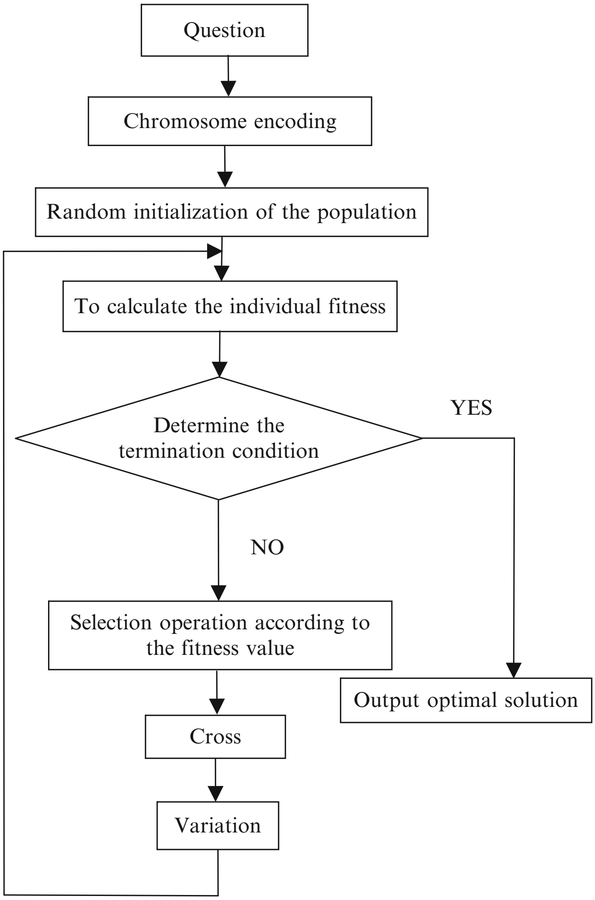

Question
Chromosome encoding
Random initialization of the population
To calculate the individual fitness
Determine the termination condition
YES
NO
Selection operation according to the fitness value
Output optimal solution
Cross
Variation

Fig. 1 GA operational processes

(6) Carry out the cross procedure to the selected individuals;
(7) Carry out the variation procedure;
(8) Return to procedure (4).

3 Achieve a Settlement for JSSP Based on GA

3.1 Process-Based Chromosome Encoding

To constructing GA for JSSP, it is an important theme that there is a suitable expression for solution. That is a suitable chromosome coding. An appropriate chromosome coding is the key step that impact GA achieving other sub-processes (Cavalieri et al. 2012). JSSP has multiple chromosome coding. Among them,

Table 1 Interpretation of the chromosome

Gene	Process	Gene	Process
4	(4,1,3,25)	1	(1,2,4,6)
4	(4,2,1,8)	2	(2,3,1,7)
3	(3,1,3,24)	2	(2,4,3,9)
2	(2,1,2,12)	4	(4,4,2,6)
4	(4,3,4,12)	3	(3,3,1,10)
2	(2,2,4,5)	1	(1,3,3,8)
1	(1,1,2,3)	1	(1,4,1,8)
3	(3,2,2,20)	3	(3,4,4,23)

process-based encoding can be the most direct expression of the correspondence between the parts and the machine as well as the whole process of scheduling, so we will adopt this way (Terzi and Cavalieri 2004).

In order to present the process-based encoding idea more clear and intuitive, this article will take a specific instance for example. Suppose a job shop scheduling problem with four parts and four machineries. A chromosome contains $4 \times 4 = 16$ genes. Each piece will appear four times in the chromosome. Each gene in the chromosomal gene cluster expresses the dependencies of procedures before and after of each part, but not a specific procedure of the part. In that case, an arbitrary chromosome with any gene sequencing can represent a feasible scheduling solution (Agrawal et al. 2012).

Part 1: (1,1,2,3) (1,2,4,6) (1,3,3,8) (1,4,1,8)
Part 2: (2,1,2,12) (2,2,4,5) (2,3,1,7) (2,4,3,9);
Part 3:(3,1,3,24) (3,2,2,20) (3,3,1,10) (3,4,4,23);
Part 4: (4,1,3,25) (4,2,1,8) (4,3,4,12) (4,4,2,6).

The first number in parentheses indicates the part's code number, the second number is the process ID of the part, the third number indicates the machine code that deal with the procedure of the part. The fourth number indicates the time required to complete the processing.

The chromosome corresponding to the hypothetical question is made up by four 1, four 2, four 3 and four 4. That is a 16-digit numeric string, for example, 4432421312243113. The explanation on this chromosome as is shown in Table 1.

3.2 Generation and Initialization of the Population

Having determined chromosome encoding rules, then adds the genes individually to the chromosome by selecting the step sequence randomly. After 16 cycles, a chromosome comes into being. Suppose that the selected group size is N = 50, then cycle the above process 50 times to form an initial population of 50 individuals.

As individual chromosomes generated by the process-based encoding rules are feasible schedules in practice, so the population is also a feasible solution set that does not need to be modified.

3.3 The Selection of the Fitness Function

The fitness value implies the degree of individuals in the population adapt to the environment. According to the principle of natural selection, the higher is that an individual's fitness value, the greater the likelihood of its survival to reproduce the next generation (Cavalieri and Pezzotta 2012). Apply this principle to GA, the fitness value will be used as the standard to choose the individual in genetic manipulation, the individual with higher fitness value will be selected by greater probability. Based on this idea, the fitness value is a probability value, which means the size of probability by which the individual will be selected.

There is no set of rules for the selection of fitness function. Generally it needs to combine the solution of the problem required to determine (Legnani et al. 2009). Then establish some connection between the fitness function and the objective function based on the model we constructed above to construct a more appropriate fitness function may well be a better method.

Define the objective function as $f(x)$. If the problem is to maximize the value of the objective function, the fitness function is $Fit[f(x)] = f(x)$; if the problem is to minimize the objective function value, the fitness function is $Fit[f(x)] = 1/f(x)$. The above scheduling model established for JSSP selects to minimize the maximum completion time as the objective function value, so it uses a simpler way of direct conversion. The fitness function value is the reciprocal of objective function values, as shown in type (1):

$$Fit(f) = 1/f = 1/Min \max_{1\leq j\leq m}\left\{\max_{1\leq i\leq n} c_{ij}\right\} \tag{1}$$

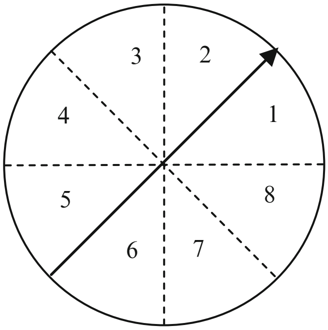
In type (1), the c_{ij} is completion time.

According to the fitness function above, the fitness values of individuals in each generation can be calculated as the basis of genetic selection operation.

3.4 The Choice of Individuals in the Population

The role of selection is to reduce the effective gene losses caused by genetic manipulation, so that individuals with high fitness can have a greater probability to survive, thus improve the convergence of the algorithm and computational efficiency on the whole (Silvestri et al. 2003). The roulette method is the most famous in all the

Fig. 2 Roulette method diagram



selection methods of GA. It is also known as proportional selection method. Here we will choose the roulette method to realize the individual choice. Roulette method is based on the basic principle that it is the ratio an individual fitness value account in the sum of all individuals fitness value in the population that determine whether the individual can be selected. It's like a lottery by turning a dial. Each turning an individual is selected (Pezzotta et al. 2012). Schematic diagram of Roulette method is shown in Fig. 2.

Using mathematical expression to represent the roulette method, the probability of individuals being selected is as shown in formula (2):

$$P_i = Fit(f)_i \Big/ \sum_{i=1}^{N} Fit(f)_i \quad i = 1, 2, \ldots, N \tag{2}$$

In the formula above, N is the size of the population, the formula above means the proportion of the individual's fitness value accounts for in the sum of the adaptation values in the entire group, obviously, the higher adaptation value chromosome instance have the greater probability of being selected.

In the selection process for instance, there is an important control parameter, called the gap which is expressed in G. That requires to be specially described here. The gap is used to control the proportion that individuals are replaced in each generation groups. For example, if G = 50 %, there will be half of the population to be replaced. If G = 100 %, all the individuals will be replaced. Here select G = 100 % to solve the JSSP.

3.5 Cross

The GA by crossover operation to generate new individual, so the cross is the core of the GA operation, the crossover operation is also known as genetic recombination (Kristianto et al. 2011). Crossover operator in a chromosome pair for execution, a combination of the two individual characteristics of the chromosome pairs to produce offspring, the offspring will be able to part or all inherited from the parent individual characteristics and effective gene. In solving the JSSP model, it need coding technique adopted by the combined problem when choose the method of crossover operation in GA. Due to the special nature of the encoding, if simply select the GA one-point crossover operation to optimize this scheduling scheme, it will be possible to produce offspring chromosome moot string encoding, decoding will is not feasible scheduling scheme. With the example of above theoretical questions to specify this cross method, as follows:

Randomly select two chromosomes as the parent generation, denoted as P1 and P2, randomly generated an integer b, which is less than total JSSP artifacts n in the model, arbitrary choice of n work piece number, pick out all of the number of genes from the two chromosomes contain, retain their relative position in the original chromosome is changeless, form two substrings, denoted as T1 and T2, the original chromosome has some empty position, and keep the position, then insert the T2 space in P1, insert the T1 space in P2, randomly generates an integer k, make sure that it is smaller than the number of genes in the chromosome, then put the P1 and P2 respectively k gene position move to the left and to the right, to generate two offspring chromosomes C1 and C2, the same relative position of the unselected gene is maintained in the original chromosome. Obviously, such a cross pattern generated chromosome is a feasible schedule.

Combining with the above assumes instance, specify the concrete application of cross method. Assuming two parent chromosomes are as follows:

P1: 4 4 3 2 4 2 1 3 1 2 2 4 3 1 1 3
P2: 2 1 1 3 2 1 4 4 2 4 3 3 2 3 4 1

Randomly selected $b = 2$, and select the work piece 1 and 2, take out 1 and 2 gene from the chromosome, obtain T1 and T2:

T1: 2 2 1 1 2 2 1 1
T2: 2 1 1 2 1 2 2 1

The parts Number 1 and 2 in the parent individuals' positions are empty:

P1: 4 4 3 - 4 - 3 - - - 4 3 - - 3
P2: - - - 3 - 4 4 - 4 3 3 - 3 4 -

Inserted T2 into the P1, and inserted T1 into the P2, randomly selected $k = 2$, respectively move the P1 and P2 two mobile location to the left and right, to get two new offspring C1 and C2:

C1: 3 2 4 1 1 3 2 1 2 4 3 2 1 3 4 4
C2: 4 1 2 2 1 3 1 2 4 4 2 4 3 3 1 3

As well as select operation, cross process also has an important control parameter, crossover probability P_c, It represents the crossover frequency, that there will be $P_c \times p$ individuals to be cross-group cross, P_c too high will result in premature convergence of the search process, is too low will make the stagnant, General admission $P_c = 0.6 \sim 0.9$.

3.6 Variation

GA's search process can be divided into global search and local search. Global search capability is mainly determined by the crossover operation. Global search presents the breadth of the search. Local search capability is mainly determined by the variation. Local search presents the depth of the search (Cavalieri et al. 2004). Although variation is an auxiliary method of genetic manipulation, it is also essential. The mutation algorithm maintains the diversity of the population to prevent the groups' premature convergence phenomenon. Crossover and mutation cooperate with each other reasonably; making the genetic algorithm is able to complete the search and optimizing process better.

Due to the special nature of the genetic code in the JSSP, the general sense of the mutation may lead to new individuals' illegality. So it adopts the inversion mutation method here. For example, in the following parent chromosomes selected two locations of genes randomly, if the code names of these two genes are different, then exchange their position to get to new chromosome individuals.

C: 4 4 2 2 1 $\underline{\mathbf{3}}$ 1 2 4 4 $\underline{\mathbf{2}}$ 1 3 3 1 3

Selected 3 and 2 and exchange their positions to get the new C′.

C′: 4 4 2 2 1 $\underline{\mathbf{2}}$ 1 2 4 4 $\underline{\mathbf{3}}$ 1 3 3 1 3

In variation, the mutation probability P_m is an important control parameter. In the new population, every gene of each instance changes by P_m. There will be $P_m \times p \times l$ times variation that take place in each generation population. Symbol l is the length of the individual. Generally, P_m has smaller value, usually being $P_m = 0.001 \sim 0.1$.

3.7 Algorithm Termination Condition

In general, in order to solve a problem, at the beginning of the genetic algorithm designing, unlimited searching down is not allowed, although the algorithm eventually convergence with probability 1 (Cavalieri et al. 2003). The convergence in theory is difficult to achieve in the actual operation, so it is often necessary to select a specific approximate convergence criterion in the design process of GA, to make a determination whether call off the search process. The most commonly used convergence criterion is to set the maximum evolution algebra in advance for the search process, when the number of iterations reaches the maximum evolution algebra, terminate the search process. Output the result. This article will choose this method as the termination condition of the algorithm.

The above are the operating principles and processes of GA applied to JSSP. Programme the above process with MATLAB to achieve the solution of the model. The processing time of the optimal scheduling solution for the example above is 85 s; the Gantt chart of the scheduling is shown in Fig. 3.

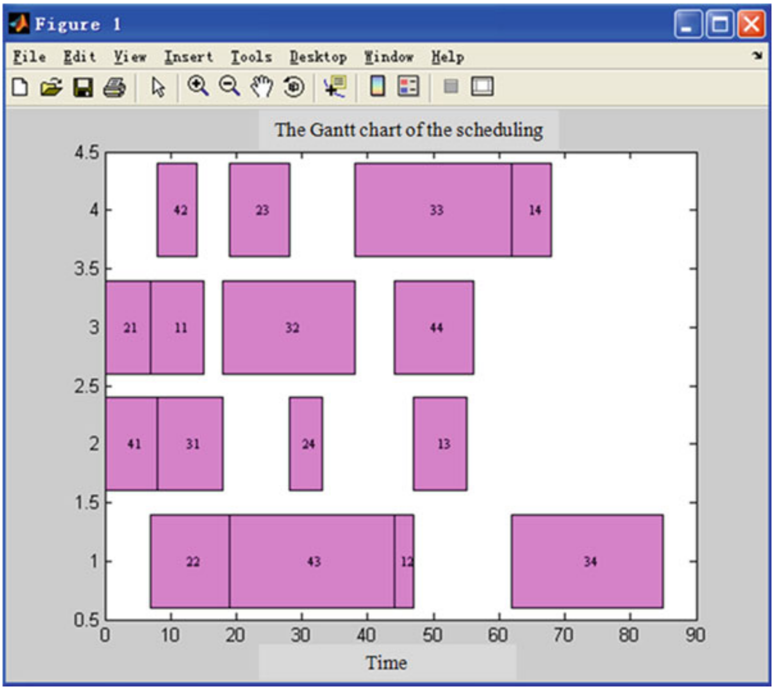


Fig. 3 The result of the scheduling Gantt chart

4 Conclusion

This article uses GA to achieve the solution to job shop scheduling problem. The solving process of the algorithm is realized by MATLAB. At last output Gantt chart of the results of JSSP to prove the feasibility and effectiveness of the model and algorithm.

References

Agrawal R, Pattanaik LN, Kumar S (2012) Scheduling of a flexible job-shop using a multi-objective genetic algorithm. J Adv Manag Res 9(2):61–74

Cavalieri S, Pezzotta G (2012) Product-service systems engineering: state of the art and research challenges. Comput Ind 63(4):278–288

Cavalieri S, Macchi M, Valckenaers P (2003) Benchmarking the performance of manufacturing control systems: design principles for a web-based simulated test bed. J Intell Manuf 14(1):43–58

Cavalieri S, Maccarrone P, Pinto R (2004) Parametric vs. neural network models for the estimation of production costs: a case study in the automotive industry. Int J Prod Econ 91(2):165–177

Cavalieri S, Pezzotta G, Shimomura Y (2012) Product-service system engineering: from theory to industrial applications. Comput Ind 63(4):275–277

Fuxiang Zhang, Wenzhong Li, Bao Jin (2009) Manufacturing resource planning of bearing manufacture enterprise based on the theory of constraint (in Chinese). Mach Tool Hydraul 37(12):39–41, 84

Guohui Zhang, Liang Gao, Peigen Li (2009) Improved genetic algorithm for the flexible job-shop scheduling problem (in Chinese). J Mech Eng 45(7):145–151

Junqiang Wang, Cuilin Zhang, Shudong Sun, Ping Fu (2005) A comparative study of production planning management and control among MRPII, TOC, and JIT (in Chinese). Manuf Autom 27(2):9–13, 29

Kristianto Y, Ajmal MM, Helo P (2011) Advanced planning and scheduling with collaboration processes in agile supply and demand networks. Bus Process Manag J 17(1):113–124

Legnani E, Cavalieri S, Ierace S (2009) A framework for the configuration of after-sales service processes. Prod Plann Control 20(2):113–124

Pezzotta G, Cavalieri S, Gaiardelli P (2012) A spiral process model to engineer a product service system: an explorative analysis through case studies. CIRP J Manuf Sci Technol 5(3):214–225

Silvestri L, Brown HR, Carra S (2003) Chain entanglements and fracture energy in interfaces between immiscible polymers. J Chem Phys 119(15):8140–8149

Terzi S, Cavalieri S (2004) Simulation in the supply chain context: a survey. Comput Ind 53(1):3–16

Tienan Zhang, Bing Han, Bo Yu (2011) Flexible job-shop scheduling optimization based on improved genetic algorithm (in Chinese). Syst Eng-Theory Pract 31(3):505–511

Ying Su, Jie Li, Huihui Yin (2010) Improved genetic algorithm for the job-shop scheduling problem (in Chinese). Manuf Technol Mach Tool 60(10):104–108