An Improved Genetic Algorithm for Solving the RGV Shop Scheduling Problem

Hao FAN¹,², Xiaoyan YU¹, De-qin SHU¹*, Wei-hong GUAN¹, Zhong-hao GUO¹, Shuo FENG¹

¹College of Information Science & Engineering, Shandong Agricultural University, Tai’an 271018, China
²Shandong Technology and Engineering Center for Digital Agriculture
*Corresponding author: De-qin SHU, shudeqin@126.com, fanhao@sda.edu.cn

Abstract. For the RGV (Rail Guide Vehicle) Shop Scheduling Problem, it is necessary to consider the RGV job scheduling under the condition of different processes with or without faults, and give the optimal solution. In this paper, the characteristics of the RGV job scheduling under the condition of different processes with or without faults are analysed and an improved genetic algorithm is proposed. The algorithm is improved in initial population selection, individual objective function, adaptive function, selection and mutation operations and so on. The improved algorithm improves the convergence speed of the initial population to the optimal solution, enables the fitness function to select the optimal individual and avoids the problems of repeated solutions and local optimum solutions in cross-selection. The improved genetic algorithm and the traditional genetic algorithm are implemented. The experimental results show that under the condition of no fault and fault in the process, the maximum number of materials processed by the improved genetic algorithm is much higher than the traditional genetic algorithm. And the time efficiency of the improved algorithm is also better than the traditional genetic algorithm. The RGV Shop Scheduling Problem can be solved more effectively by the improved genetic algorithm.

1. Introduction
With the development of information technology, the processing system of industrial integration and intelligent integration is becoming more and more popular. In most automated workshop systems, there will be one or more Rail Guide Vehicles (RGV). However, there are many uncertain factors in the actual processing process, such as uneven material, temperature, tool wear, machine failure and so on, which will make the intelligent RGV unable to achieve theoretical optimization problems. Therefore, improving the scheduling algorithm and finding the optimal solution in scheduling is the key to the scheduling of intelligent RGV [1-3].

In practice, genetic algorithm, simulated annealing algorithm, tabu search algorithm and ant colony algorithm are often used for job scheduling problems [4-6]. Among them, the concept of genetic algorithm was first proposed by Professor John Holland of Michigan University in 1975[7]. It is often used because of its excellent computational performance and remarkable application effect. However, the traditional genetic algorithm may have premature convergence or slow convergence in solving the optimal solution problem, which leads to the emergence of suboptimal solution [8-10].

In this paper, an improved genetic algorithm is designed for RGV job shop scheduling problem. The algorithm obtains the initial population through the greedy strategy, uses the single job crossover...
to achieve the crossover, and performs the selective crossover according to the fitness value during the
cross operation, avoiding the problems of the repeated solution and the local optimal solution. The
improved algorithm is implemented by C++ and compared with the traditional genetic algorithm.
Compared with the traditional genetic algorithm, the improved algorithm can not only improve the
convergence speed of the initial population to the optimal solution, but also further improve the time
efficiency. Therefore, there is a more optimized effect in the actual job scheduling process with the
improved algorithm.

2. Problem analysis and description
RGV shop scheduling problem (shown in figure 1) means that RGV can automatically control the
moving direction and distance according to the command, grab the material from the material
conveyor, and then complete the loading and unloading and cleaning of the CNC (Computer
Numerical Controller). Wait for job tasks, where each CNC occurs an even number of times. There are
two basic material processing situations in RGV scheduling. The maximum numbers of materials that
can be processed in 8 hours in the case of process-free material processing. The maximum number of
materials that can be processed in 8 hours in the case of material processing with 1% failure rate in the
process.

Figure 1. Diagram of RGV job shop scheduling system.

The form of the machining job in the shop scheduling problem can be described in the form of a
table, as shown in table 1.

Table 1. Three sets of job parameters in job shop scheduling.

| Processing parameters                           | Group1 | Group2 | Group3 |
|------------------------------------------------|--------|--------|--------|
| Time for RGV to move 1 unit                    | 20     | 23     | 18     |
| Time for RGV to move 2 unit                    | 33     | 41     | 32     |
| Time for RGV to move 3 unit                    | 46     | 59     | 46     |
| The time for CNC to complete one process       | 560    | 580    | 545    |
| First Step Time of CNC Two-step Material Processing | 400   | 280    | 455    |
| Second Step Time of CNC Two-step Material Processing | 378   | 500    | 182    |
| The time for RGV unload once for CNC1#, 3#, 5#, 7# | 28     | 30     | 27     |
| The time for RGV unload once for CNC2#, 4#, 6#, 8# | 31     | 35     | 32     |
| The cleaning time of RGV for a material        | 25     | 30     | 25     |

3. Traditional genetic algorithm
The traditional Genetic Algorithm (GA) [7], proposed by Professor John Holland, is a meta-heuristic
search algorithm which simulates the natural selection of Darwin's biological evolution theory and the
computational model of the biological evolution process of genetic mechanism [11]. The genetic
algorithm draws on the evolutionary law of “survival of the fittest, survival of the fittest”. Starting
from a population that represents the potential solution set of the problem, evolutionary and better approximate solutions is evolved from generation to generation. In each generation, according to the fitness of individuals in the problem domain, individuals who perform well are selected and combined with genetic operators to generate crossovers and mutations to produce populations that represent new solutions. The genetic algorithm generally consists of four parts: encoding and decoding, fitness function, genetic operator and control parameters. The choice of each parameter has an important influence on the convergence of the algorithm and the solution to the problem. The advantage of genetic algorithm in solving scheduling problem is that it can randomly move from one job to another, which makes the algorithm not limited to local optimal solution in the search process. It is suitable for parallel processing, robust, simple and universal, and search [12, 13]. The flow of the genetic algorithm is shown in Figure 2.

![Figure 2. Process of genetic algorithm.](image)

4. Improved genetic algorithm
The improved genetic algorithm can effectively solve the scheduling problem of the workshop RGV, and make up for the shortcomings of the traditional genetic algorithm proposed by Professor John Holland. The methods proposed in this paper have made appropriate improvements in the initial population selection, individual objective function, adaptation function, selection as well as the mutation operations, and the specific improvements are as follows.

The improved genetic algorithm adopts the greedy strategy for the selection of the initial population for the traditional genetic algorithm, avoiding the instability and time-consuming shortcomings of the population generated by the traditional random method, and improving the speed at which the initial population converges to the optimal solution. Considering that there is a parity number problem when RGV schedules CNC, when defining the individual objective function, the time objective function with position coordinates of RGV is adopted, which makes the representation of CNC scheduling more accurate. When determining the adaptive function, it is not simply the individual objective function as an adaptive function. Instead, the objective function with parameters is used, so that the fitness function can select the optimal individual to get the optimal solution. The
improved genetic algorithm performs selective crossover according to the fitness value during the crossover operation, compared with traditional genetic algorithms, crossover can produce more optimal solutions, and avoids the problem of repeated solutions and local optimal solutions. In the mutation operation, the improved genetic algorithm adopts a method of randomly changing the order of the CNC, so that the probability of occurrence of the optimal solution increases. The specific steps to improve the genetic algorithm are described in detail in section 4.1-4.6.

4.1. Coding method
Since the genetic algorithm cannot directly deal with the feasible solution in the practical application problem, so the solution of the problem is converted into a "chromosome" in which each gene is a single individual. Using symbolic coding methods, the actual abstraction problem is transformed into a mathematical problem that is easy to solve.

There are a total of 8 CNC in the workshop. Each CNC is a gene of chromosome, which is represented by the symbol set \{0, 1, 2, 3...7\}. The order in which the genes are arranged in the chromosome is the order in which each CNC is used in this feasible solution. For example, if there are n kinds of dispatch routes, the n coded routes can be expressed as formula (1).

\[
\begin{cases}
0,1,2,3,4,5,6,7,0,1,2,3,5,6,4,7 \\
4,2,3,1,5,6,7,0,4,0,2,3,1,5,6,7
\end{cases}
\]

(1)

When the processed material is n, the code length is as formula (2).

\[
\sum_{i=0}^{n} n_i \times 2
\]

(2)

Where \(n_i\) is the length of the encoded route when the number of materials is \(i\).

4.2. Initial population selection
In the case of a certain number of genetic operators and iterations, the randomly generated initial population has a great influence on the results and global convergence. If improperly selected, the calculation results will be unstable. In this paper, the greedy strategy is used to construct the initial solution of the population, and the shortest path to the CNC by RGV is chosen as the greedy criterion, and a set of better initial feasible solutions is obtained. The size of the population represents the size of the initial solution set.

Too small is easy to lose the diversity of the population, resulting in the inability to find the optimal solution; too much will increase the amount of calculation, therefore, a balance needs to be made between the two methods.

The size of the initial population can be selected according to subjective experience. If the base of the population is larger, the diversity of the population will be enhanced and the result will be more reliable. However, if the population base is set too large, the convergence speed of the algorithm will be too slow. Hence, according to the empirical value, the size \(C\) of the population of this paper is chosen to be 20.

4.3. Individual objective function
The material processing operation of one process can be regarded as the scheduling of the CNC which is in no order. All CNC can process this process, and what we want is the shortest sequence in the case of different CNC scheduling, that is, the shortest time sequence. Constraint condition is described as formula (3).

\[
S_m = \begin{cases} 
  c_2, & p \% 2 = 0 \\
  c_1, & p \% 2 \neq 0 
\end{cases}
\]

(3)

Among them, \(S_m\) indicates the time required for RGV to load and unload different CNCs. Because RGV gives the CNC with odd and even numbers a different time for loading and unloading, \(S_m\) needs to be discussed separately, when the CNC number is 1#, 3#, 5#, 7#, the time is taken as \(c_1\).
When the CNC number is 2#, 4#, 6#, 8#, and the time is taken as \( c_2 \), so the time objective function obtained last described as formula (4).

\[
t_s = \left| \frac{p_{\text{pos}} - p}{2} \right| * t_i + S_m
\]

(4)

Where \( t(x) \) is the final total time, \( p \) is the position coordinate of the CNC, \( p_{\text{pos}} \) is the position coordinate of RGV, \( t_i \) is the time required for one unit of RGV motion, \( S_m \) is the shortest time series.

4.4. Determination of adaptive function

Using the objective function transformation method with parameters, the objective function is minimized when the objective function \( t(x) \) is converted to the fitness function \( g(x) \) is described as formula (5).

\[
g(x) = \begin{cases} 
C_{\text{max}} - t(x), & t(x) < C_{\text{max}} \\
0, & \text{otherwise}
\end{cases}
\]

(5)

Among them, \( C_{\text{max}} \) takes the maximum value of all algebras so far, and \( t(x) \) is the objective function of time.

4.5. Choice and crossover

In terms of the selection operation of the genetic algorithm, selection refers to selecting an individual in the parent population to perform subsequent crossover operations. In the algorithm, not every individual has the same probability to enter the next generation. For the scheduling problem of RGV in this paper, the choice of CNC is based on the method of proportional selection under the premise of the objective function, and the formula for calculating the individual selection probability is as formula (6).

\[
s_{\text{sum}}(i) = \frac{\text{up}(i)}{\sum_{i=1}^{N} \text{up}(i)}
\]

(6)

Among them, \( \text{sum}(i) \) is the probability of selection of individual \( i \), \( \text{up}(i) \) is the fitness value of CNC individual \( i \), and \( N \) is the number of individuals in the population. Obviously, \( \text{up}(i) \) reflects the proportion of individual \( i \)'s fitness in the total fitness of the entire population. The probability of CNC individuals being selected varies with the fitness value. After obtaining the selection probability of each individual, this paper uses the method of roulette to select, the probability of each individual is simultaneously expanded several times, and a number of individuals are selected using a random integer method. For each of the eight CNC individuals, the probability of each CNC individual being selected (expanded by 10 times) is 8, 7, 6, 5, 4, 3, 2, 1, respectively. The cumulative probability of each CNC individual is shown in Table 2.

| Individual | 1# | 2# | 3# | 4# | 5# | 6# | 7# | 8# |
|------------|----|----|----|----|----|----|----|----|
| Selection probability | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| Cumulative probability | 36 | 28 | 21 | 16 | 12 | 9 | 7 | 5 |

After calculating the cumulative probability, set a zero point on the disc, starting from the zero point, and labeling the calculated individual cumulative probability value on the circular turntable, as shown in figure 3.
The crossover operation in the genetic algorithm refers to the process in which two pairs of chromosomes exchange some of their genes in a certain way, thereby increasing the number of two new individuals, by cross-operation, a large number of new individuals are generated, which can not only inherit the good genes, but also can increase the diversity of the population [14]. Each time a random number rand is generated before the crossover, it represents the position of the crossover, and then the gene is exchanged at this position. Common cross-over methods are single-point crossover, two-point crossover, multi-point crossover, and uniform crossover [15-16]. The crossover criterion is to make the crossover probability range from 0.75 to 0.95, but due to the different importance of each gene locus, this traditional genetic algorithm based on probabilistic cross-operation makes the emergence of the optimal solution very uncertain and cannot effectively search for the optimal solution in the optimization space. Therefore, this paper adopts an improved crossover method for the scheduling problem of RGV, that is, a crossover is set for two CNC scheduling sequences to cross, but the crossover of the crossover points is recorded to record the fitness values of two new individuals \((g_1, g_2)\). Then, the maximum fitness value \(\max(g_1, g_2)\) is selected, and the maximum fitness value of the individual is compared with the maximum fitness value of the parent, if the new individual fitness value is greater than the parental fitness value, the crossover is retained, if the new individual fitness value is smaller than the parent fitness value, the parent is directly copied to the next generation. The specific crossover process is as follows:

Sequence A: \([1 2 3 4 5 6] \rightarrow [1 2 3 1 5 6]\)
Sequence B: \([4 5 6 1 2 3] \rightarrow [4 5 6 4 2 3]\)

4.6. Variation
Mutation operation is an important factor in genetic algorithm. It can be used to generate new genes. Unlike crossover operations, two individual exchange genes are required. The mutation operation can directly replace some genes of the chromosome with other alleles to form a brand new individual and achieve local optimization. When the algorithm falls into a stagnant state too early, the mutation can cause the algorithm to jump out of a stagnant state. Individual mutation has a certain probability, which is represented by \(P_2\). The value of the probability of mutation is also obtained through multiple experiments. The general recommendation is 0.001-0.1.

In order to increase the probability of the optimal solution, this paper adopts the following method. After a CNC scheduling sequence occurs, the mutated method is used to change the order of any CNC, and another CNC switching sequence is obtained, which finally makes the optimal. The probability of solution increases. The specific process is as follows:

Sequence A: \([1 2 3 4 5 6 7 8] \rightarrow [1 2 3 6 5 4 7 8]\)
Sequence B: \([1 2 3 4 5 6 7 8] \rightarrow [1 2 8 6 5 4 7 8]\)

5. Experimental results and analysis
Aiming at RGV Job Shop Scheduling problem, the improved genetic algorithm and the traditional genetic algorithm proposed by Professor John Holland are implemented respectively, and the maximum number of processed materials and time efficiency are compared in detail.
5.1. Comparison of the maximum number of processed materials

The experimental results are compared as table 3.

| Situation | Material Quantity in one Procedure | Material Quantity in a Faulty Procedure |
|-----------|------------------------------------|----------------------------------------|
| Group1    | 313                                | 309                                    |
| Group2    | 300                                | 294                                    |
| Group3    | 320                                | 314                                    |

As can be seen from the data in Table 3, the final result of the improved genetic algorithm is that the maximum number of materials that can be processed in 8 hours in one process is 313, 300 and 320, and the maximum number of materials that can be processed in 8 hours in the case of a fault in one process is 309, 294 and 314.

| Situation | Material Quantity in one Procedure | Material Quantity in a Faulty Procedure |
|-----------|------------------------------------|----------------------------------------|
| Group1    | 150                                | 143                                    |
| Group2    | 146                                | 136                                    |
| Group3    | 155                                | 150                                    |

As can be seen from the data in Table 4, the final result of the traditional genetic algorithm is that the maximum number of materials that can be processed in 8 hours in one process is 150, 146, 155, and the maximum number of materials that can be processed in 8 hours in the case of a fault in one process is 143, 136 and 150. Data fitting analysis was performed as shown in figure 4 and figure 5. The abscissa represents the number of test data sets, and the ordinate represents the number of processed materials.

![Figure 4](image1.png)

**Figure 4.** Material quantity data fitting in one process.

![Figure 5](image2.png)

**Figure 5.** Material volume data fitting for fault handling in a process.

By comparing the data in Tables 3 and 4, it is concluded that the average number of materials that can be processed by improved genetic algorithm and traditional genetic algorithm in 8 hours without machine failure is 311 and 150 respectively, and the average number of materials that can be
processed in 8 hours with machine failure is 302 and 143 respectively. Therefore, the improved genetic algorithm is almost doubled in the number of materials processed compared to the traditional genetic algorithm.

5.2. Comparisons of execution time
The time efficiency of these two algorithms is compared on a personal computer with Intel (R) Celeron (R) 2957U @ 1.40GHz processor and 8.0GB memory capacity. The results are as follows:

| Table 5. Comparisons of execution time between two algorithms for the first set of data. |
|---------------------------------|---------------------------------|--------------------------|
| Number of times | Traditional genetic algorithm time /ms | Improved genetic algorithm time /ms |
|-----------------|---------------------------------|--------------------------|
| First time      | 1702                            | 1517                     |
| Second time     | 1624                            | 1533                     |
| Third time      | 1611                            | 1542                     |

| Table 6. Comparisons of execution time between two algorithms for the second set of data. |
|---------------------------------|---------------------------------|--------------------------|
| Number of times | Traditional genetic algorithm time /ms | Improved genetic algorithm time /ms |
|-----------------|---------------------------------|--------------------------|
| First time      | 1660                            | 1531                     |
| Second time     | 1625                            | 1489                     |
| Third time      | 1645                            | 1582                     |

Figure 6. Comparing the execution time before and after the improvement of the algorithm.

It can be seen from the data in Figure 6 that the improved genetic algorithm is faster and more efficient than the traditional genetic algorithm.

In summary, when dealing with the shop scheduling problem, the maximum number of materials that can be processed by the improved genetic algorithm is much higher than the traditional genetic algorithm. And the time efficiency of the improved algorithm is also better than that of the traditional genetic algorithm. The improved genetic algorithm is very suitable for solving the RGV shop scheduling problem, and it can improve the efficiency of shop job scheduling.

6. Conclusion
In this paper, the Shop scheduling problem is comprehensively analysed. And an improved genetic algorithm is used to solve the job scheduling problem under the condition of different processes. In the process of initial population generation, the greedy strategy is adopted to generate better initial feasible solution, which makes it easy to get the optimal solution. And in the operation of determining the individual fitness function, the adaptive function with parameters is adopted to make the selection of individuals more suitable to the actual situation. Finally, for the improvement of the crossover operation, the crossover point is no longer selected randomly, but the fitness of the new individual is used to judge whether to retain the crossover. Eventually, the problem of premature convergence or slow convergence of the optimal solution is avoided.

It can be seen from the experimental comparison that compared with the traditional genetic algorithm, the improved genetic algorithm not only greatly increases the number of processed materials, but also has a higher time efficiency.
The improved genetic algorithm proposed in this paper is validated by calculating the number of materials and running time of the algorithm through C++ and MATLAB simulation, providing a method and approach for the research of RGV job scheduling problem. At the same time, the algorithm has good portability and can be easily applied to structural design, artificial intelligence and complex system analysis in engineering.

Acknowledgements
This work is supported by Natural Science Foundation of Shandong Province (ZR2016FB19, ZR2019MF026 and ZR2017MD018), National Natural Science Foundation of China (41471299) and Technology Innovation Fund Project of Shandong Agricultural University (23659).

References
[1] Jiang, W., Fei, H.E., Tong, Y., et al. (2016) Hybrid algorithm for rounding rail guided vehicle optimization scheduling. Computer Engineering & Applications., 52(22): 242-247.
[2] Xie, C., Allen, T. T. (2015) Simulation and experimental design methods for job shop scheduling with material handling: a survey. The International Journal of Advanced Manufacturing Technology, 80:233-243.
[3] Yu, H., Wang, L. (2012) Genetic Algorithm Combined with Simulation for Job Shop Scheduling Problem in Mechanical Engineering. Lecture Notes in Electrical Engineering, 176:139-144.
[4] Kachitvichyanukul, V., Sitthitham, S.A. (2011) Two-stage genetic algorithm for multi-objective job shop scheduling problems. Journal of Intelligent Manufacturing, 22(3):355-365.
[5] Van Laarhoven,P. J. M., Aarts, E. H. L., Lenstra, J. K. (2017) Job shop scheduling by simulated annealing. Operations Research, 40(1):113-125.
[6] Bo, P., Lü, Z., Cheng, T. C. E. (2015) A tabu search/path relinking algorithm to solve the job shop scheduling problem. Computers & Operations Research, 53:154-164.
[7] Holland, J. (1975) Adaptation in Natural and Artificial Systems. MA: MIT Press, Cambridge. ISBN 978-0262581110.
[8] Su,N., Zhang,M., Johnston,M., et al. (2014) Automatic Design of Scheduling Policies for Dynamic Multi-objective Job Shop Scheduling via Cooperative Coevolution Genetic Programming. IEEE Transactions on Evolutionary Computation, 18:193-208.
[9] Rehab, F., Abdel-Kader. (2018) An improved PSO algorithm with genetic and neighborhood-based diversity operators for the job shop scheduling problem. Applied Artificial Intelligence, 2:1-30.
[10] Defersha,F. M., Bayat-Movahed, S. (2018) Linear programming assisted (not embedded) genetic algorithm for flexible jobshop scheduling with lot streaming. Computers & Industrial Engineering, 117:319-335.
[11] Park, B. J., Choi, H. R., Kim, H.S. (2003) A hybrid genetic algorithm for the job shop scheduling problems. In: Elsevier B.V. Munich.PP.597-613.
[12] Liu, C.Y., Zou, C.M., Wu, P. (2015) A Task Scheduling Algorithm Based on Genetic Algorithm and Ant Colony Optimization in Cloud Computing. In:International Symposium on Distributed Computing & Applications to Business. Atlanta. pp. 178-187.
[13] ZHANG, L., ZHANG, B. (2001) Good Point Set Based Genetic Algorithm. Chinese Journal of computers, 24(9):917-923. (in Chinese).
[14] Costa, A., Cappadonna, F.A., Fichera, S.J. (2017) A hybrid genetic algorithm for minimizing makespan in a flow-shop sequence-dependent group scheduling problem. https://doi.org/10.1007/s10845-015-1049-1.
[15] Aziza, H., Krichen, S. (2018) Bi-objective decision support system for task-scheduling based on genetic algorithm in cloud computing. https://doi.org/10.1017/s00607-017-0566-5
[16] Dao, T.K., Pan, T.S., Nguyen, T.T., et al. (2018) Parallel bat algorithm for optimizing makespan in job shop scheduling problems. Journal of Intelligent Manufacturing, 29(2):1-12.