Dynamic Scheduling Strategy of RGV

Wen Fu[1, a], Qingchang Wang[2, b], Yunzhao Li[3, c], Heng Zhang[4, d], Fei Wang[5, e], and Huan Li[6, f]

[1]Marine Electrical Engineering, Dalian Maritime University, Dalian 116026, China.

[2]School of Maritime Economics and Management, Dalian Maritime University, Dalian 116026, China.

[3]Department of Physical Education, Dalian Maritime University, Dalian 116026, China.

[4]Naval Architecture and Ocean Engineering College, Dalian Maritime University, Dalian 116026, China.

[5]Information Science and Technology College, Dalian Maritime University, Dalian 116026, China.

[6]School of Navigation College, Dalian Maritime University, Dalian 116026, China.

[a]18742524075@163.com, [b]Wangqingchang666@163.com, [c]Chinaliyunzhao1018@163.com, [d]2280286252@qq.com, [e]yarloy@dlmu.edu.cn, [f]949955708@qq.com

Keywords: Rule-based Adaptive Genetic Algorithm; Adaptive Crossover and Mutation Probability; Flexible Population

Abstract: This paper focuses on the dynamic scheduling strategy of intelligent RGV (Rail Guided Vehicle). The goal of this problem is to find an optimal scheduling scheme for processing some materials within a certain time (one shift). To make the problem easier to solve, we convert the goal of the problem into a scheduling scheme that solves the processing of n-piece materials and requires the shortest time. At the same time, according to the purpose of this problem, the shortest time is limited to one shift, and then the number of materials is slightly adjusted to make the time close to eight hours. In the process, we establish a mathematical model and adopt a rule-based adaptive genetic algorithm. To avoid obtaining a low-reasonability scheme and reduce the efficiency of the algorithm, we propose some rules for manual intervention when generating the initial population. We use the adaptive crossover and mutation probability and flexible population to replace the fixed parameters in the traditional genetic algorithm. Finally, we analyze the performance of the algorithm by MATLAB simulation experiment and the effectiveness of the algorithm is verified.

1. Introduction

With the improvement of industrial automation level, RGV has been widely used in actual processing and production. Depending on the mode of motion, RGV can be divided into a circular orbital type and a linear reciprocating type. The circular orbital RGV system is highly efficient and can work simultaneously in multiple shuttle vehicles. While the linear reciprocating type generally includes only one RGV for reciprocating motion, and the efficiency is relatively low compared with the circular RGV system. This paper mainly studies the scheduling problem of the linear RGV in an intelligent processing system .The problem can be described as follows: the existing materials that need to be processed is $Tasks = \{Task_i \mid i = 1, 2, \cdots, n\}$ .And the intelligent processing system consists of 8 CNC(Computer Number Controllers) ,1 Rail Guide Vehicle (RGV), 1 RGV linear track, 1 feeding conveyor belt and 1 blanking conveyor belt. The schematic diagram of the system is shown in Figure 1.

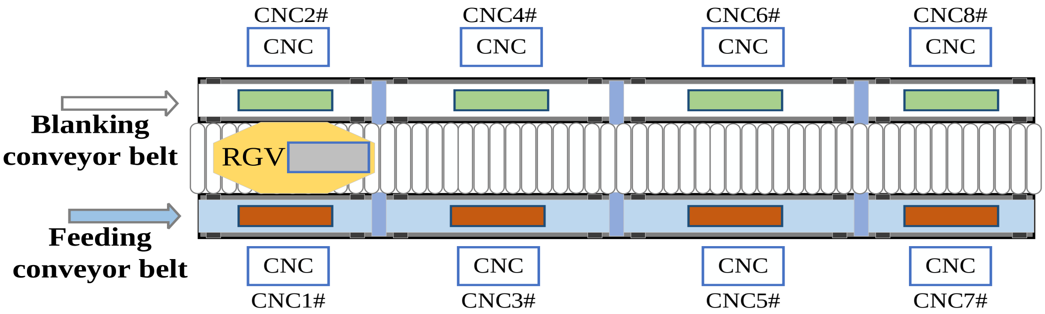


Figure 1. The schematic diagram of the intelligent processing system

To solve the problem more intuitively, we use some data of a certain factory process as a reference. And the part of data is as follows:

Table 1. The operating parameters of the processing system (Time unit: second)

Parameters	Group 1	Group 2	Group 3
The time of RGV moving 1 unit	19	22	17
The time of RGV moving 2 units	32	40	30
The time of RGV moving 3 units	45	58	45
The time required for CNC to complete a material	559	577	543
The time for RGV to complete a material's cleaning	24	28	26
The time required for RGV to load and unload at one time	28	30	27

To simplify the given problems and modify it more appropriate for simulating real-life conditions, we make the following basic hypotheses, each of which is properly justified.

- RGV is a single station, that is, the shuttle can only be responsible for one handling task at a time.
- When the RGV and the CNC have a fault during the entire dispatching cycle, the system stops running and does not affect the scheduling optimization process.
- Each CNC is idle at the initial moment when the entire system begins.
- Once each task is executed, it will not be interrupted by other tasks.

2. Scheduling Optimization Model

For this problem, we analyze the RGV timeline to list mathematical expressions to avoid increasing the complexity of the problems by analyzing multiple timelines simultaneously. However, we are not directly solving the scheduling scheme that maximizes the working efficiency of the system in a shift. We convert it into a scheduling scheme that solves the processing of n-piece materials and requires the shortest time to make the system most efficient. At the same time, according to the goal of the problem, the minimum time is guaranteed to be controlled in about 8 hours. For RGV, the time $t_{Task_i}\,(i=1,2,\cdots,n)$ required to process a piece of material is

$$t_{Task_i} = t_w + t_m + t_{ud} + t_c \tag{1}$$

Where t_w is the RGV waiting time before processing a material, t_m is the RGV moving time before processing a material, t_{ud} is the time required for RGV to complete a loading and unloading and t_c is the time spent on cleaning a material. We do not consider the dwell time of the material in the CNC separately, but the dwell time is included in the waiting time t_w before the RGV processes a material.

Therefore, for RGV, the time required to process n materials are

$$t_{Tasks} = \sum_{i=1}^{n} t_{task_i} \tag{2}$$

When all tasks are completed for the shortest time, it is the optimal solution for the RGV system scheduling problem, so the objective function is

$$\min t_{Tasks} = \sum_{i=1}^{n} t_{task_i} \tag{3}$$

3. Rule-based Adaptive Genetic Algorithm Model

When the genetic algorithm solves the scheduling problem, the random generation method is generally used to form the chromosomes and obtain the initial population, so that a variety of low-reasonability schemes may be formed before the iteration begins. Therefore, many iterations are required to obtain an optimization scheme. Not only does it reduce the computational efficiency of the algorithm, but it also tends to fall into local optimal solutions without a good scheduling scheme.

However, the rule-based scheduling can only solve the scheduling problem of the current idle RGV, and it is impossible to predict the task assignment, so that it is difficult to obtain a better solution when dealing with the large-scale scheduling problem. In this paper, we use the rule-based adaptive genetic algorithm to improve the initial convergence of the initial population acquisition scheme to obtain high-quality initial population, thus improving the convergence performance of the algorithm.

3.1 Rule-based Scheduling

Rule-based scheduling is widely used in the existing scheduling system and it is an instant scheduling strategy. The biggest advantage of rule-based scheduling is that the algorithm has low complexity and can be applied to dynamic real-time scheduling and complex large-scale scheduling. When the RGV system and the task release system are integrated, the rule scheduling can be triggered by other systems when the new task is released or the RGV’s state changes.

- Rule 1: The principle of proximity---When there are multiple CNC machines in the idle state, RGV selects the nearest CNC to complete the processing task.
- Rule 2: Non-repetition principle ---The same CNC machine cannot process two materials in a row, that is, there are no continuous and repeated genes on the same chromosome.

3.2 Adaptive Genetic Algorithm

Since the RGV’s scheduling is an immediate problem, the fixed parameters are not conducive to the solution of its scheduling problem. The traditional genetic algorithm's crossover probability and mutation probability mostly rely on empirical value, and the choice of its parameters directly affects the global optimality and convergence of the algorithm.

The adaptive genetic algorithm adopts adaptive genetic evolution parameter. When the individual's difference is large, it tries to narrow the gap, which not only allows the excellent individual to fully develop, but also gives the poor individual a certain chance of evolution; when the individual's difference is small, it maximizes the gap and can better promote the evolution of the population.

4. Encoding and Generating the Initial Population

We encode the CNC machine serial number by natural number coding to form a chromosome divided into L segments. Each chromosome corresponds to a scheduling scheme, and the serial number $g_{ij}(g_{ij} = 1, 2, \cdots, n)$ on each chromosome corresponds to the CNC machine serial number for processing a certain material. Thus, we can get the length of the chromosome

$$L = n \tag{4}$$

As mentioned earlier, in the process of randomly generating chromosomes consisting of CNC machine serial numbers, we artificially intervene based on the above rules to produce the high-quality initial population.

To obtain the optimal scheduling scheme, we determine the size of the population, that is, the number of chromosomes T (0~100) as a variable rather than a fixed parameter. Finally, we get the CNC machine serial number matrix (the initial population) diagram shown in Figure 2.

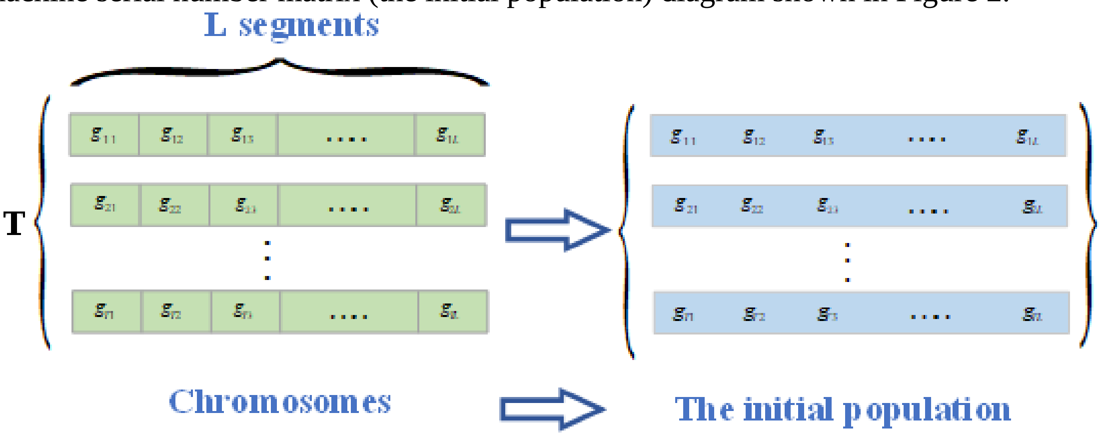


Figure 2. The CNC machine serial number matrix (the initial population)

5. Fitness Function

In the genetic algorithm, fitness refers to the ability of individual adaptation to the living environment. The higher the fitness is, the greater the chance of survival the individual will have. In this intelligent processing system, we measure the fitness in terms of the time that each scheduling scheme spends on processing the materials. The shorter the time for each scheduling scheme to process materials, the more likely it is to be selected. Therefore, the fitness of each scheduling scheme S_k can be expressed by the opposite of the objective function value:

$$eval(S_k) = f(t_{task_i}) = -\sum_{i=1}^{n} t_{task_i}, k = 1, 2, \cdots \tag{5}$$

The sum of the fitness values of all scheduling schemes is

$$F = \sum_{k=1}^{pop\_size} eval(S_k) = \sum_{k=1}^{T} eval(S_k) \tag{6}$$

The probability that each scheduling scheme is replicated is

$$P_k = \frac{eval(S_k)}{F} \tag{7}$$

The cumulative probability that each scheduling scheme is replicated is

$$Q_k = \sum_{j=1}^{k} P_k \tag{8}$$

As shown in Table 2, we obtain the fitness and probability of some scheduling schemes through the GA toolbox in MATLAB.

Table 2. The fitness and probability of some scheduling schemes

Scheduling scheme	Fitness value	P_k	Q_k
$S_1 = [1,3,4,6,8,7,6,4\cdots]$	-28945	0.111180	0.262534
$S_2 = [2,1,4,3,6,7,5,8,\cdots]$	-34508	0.165077	0.343224
$S_3 = [3,5,6,7,5,4,3,1,\cdots]$	-34654	0.124330	0.433069
$S_4 = [2,4,5,3,2,7,8,6,\cdots]$	-23784	0.057643	0.576433

6. Crossover

Among the parameters of the genetic algorithm, the crossover and mutation operators directly affect the convergence speed of the algorithm and the ability to jump out of the local optimum. In traditional genetic algorithms, fixed crossover fraction and mutation are not conducive to convergence to global optimality. Thus, we use adaptive crossover probability and mutation probability that can improve this phenomenon. The adaptive crossover fraction P_{cross} calculation formula is as follows:

$$P_{cross} = \begin{cases} k_1 \times \sin(\frac{\pi}{2} \times \frac{f_{\max} - f_k}{f_{\max} - f_{avg}}),\ f_k > f_{avg} \\ k_2,\ f_k \le f_{avg} \end{cases} \tag{9}$$

Where $f_{\max}$ is the maximum fitness value, f_{avg} is the average fitness value and f_k is the individual fitness value.

In this paper, we apply the two-point crossover method, and the crossover point is randomly selected.

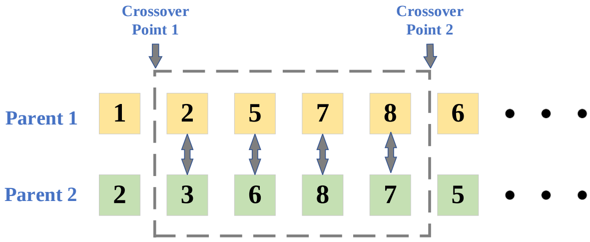


Figure 3. The schematic of crossover

7. Mutation

For the range of the genes to be mutated, we choose two kinds of mutations. When the number of tasks is large, if there are redundant tasks that are not involved in chromosome coding and the individual fitness value is less than the average fitness value, we adopt the external mutation; otherwise, we choose self-crossing mutation.

- The external mutation: Replace the mutation position gene with a gene that does not exist in the original chromosome.

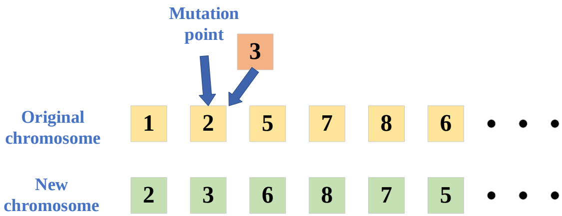


Figure 4. The schematic of the external mutation

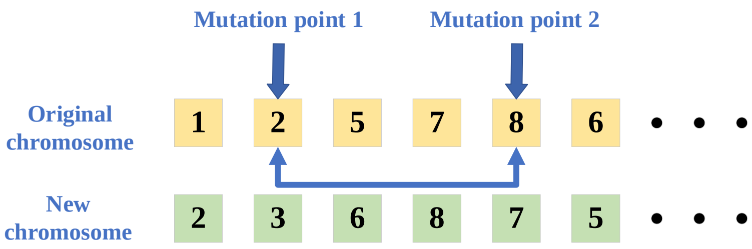


Figure 5. The schematic of the self-crossing mutation

- The self-crossing mutation: Randomly generate two mutation points to exchange genes at the mutation points.

And the adaptive mutation probability is as follows:

$$P_m = \begin{cases} k_1 \times \sin\left(\frac{\pi}{2} \times \frac{f_{\max} - f_k}{f_{\max} - f_{avg}}\right), f_k > f_{avg} \\ k_2, f_k \le f_{avg} \end{cases} \quad (10)$$

8. Conclusion

To make the whole evolution process of genetic algorithm intuitive, we calculate the fitness value of various scheduling schemes through MATLAB and draw the following three-dimensional figure.

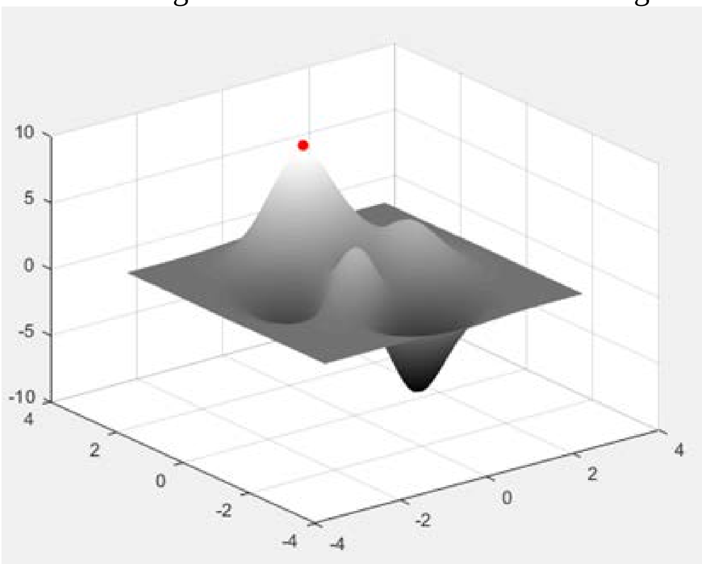
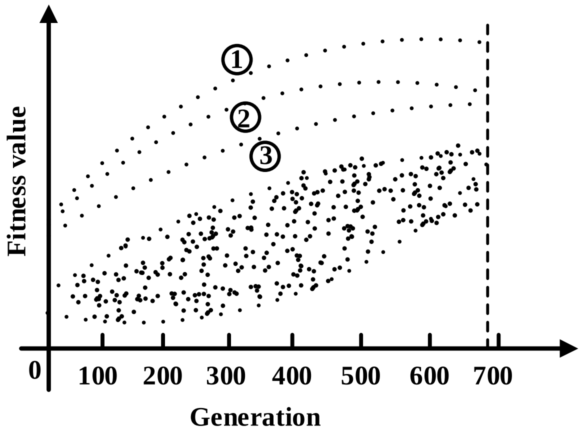


Figure 6. The 3D figure of fitness value

As we can see from the figure, there are many mountain peaks in the figure. Among them, the mountain peak is what we call the local optimal solution, and the peak of the mountain marked by the red dot is the global optimal solution. And we can find that the rule-based adaptive genetic algorithm can jump out of the local optimal solution. The target solution we need is a scheduling scheme corresponding to the vicinity of the red dot. Finally, we obtain the trend of the best individual fitness value and the average fitness value of the population during the evolution of the genetic algorithm.



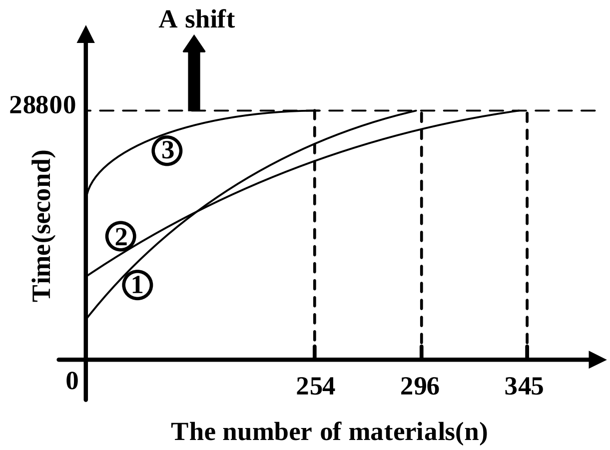


Figure 7. The change trend of the fitness value

Figure 8. The process of selecting the optimal solution

The evolution of the population to about 680 generations has stopped because the optimal population does not evolve for a long time and the algorithm automatically stops running. At the same time, we select several individuals (①,②and③) with high fitness value .Then, we calculate the processing time of the schemes corresponding to these individuals .What we find is that the processing time is less than the specified time. Thus, we continue to increase the number of materials, so that the processing time of these selected schemes approaches 8 hours. The scheduling scheme corresponding to curve ② processes the maximum number of materials (345) in 8 hours. Consequently, the optimal scheduling scheme corresponding to this curve is $\left[4,7,1,2,3,5,6,8,1,4,2,3,6,\cdots\right]$.

References

[1] Ji Shuxin, Qian Jixin, Sun Youxian. Application of Genetic Algorithm in Job Shop Scheduling [J]. Systems Engineering - Theory & Practice, 1998, 18(5): 34-39.

[2] Jiang Wei, He Fei, Tong Yifei, et al. Scheduling optimization of circular orbit RGV system based on hybrid algorithm [J]. CEA, 2016, 52(22): 242-247.

[3] Ding Xuefang,Wang Qi.Wi-Fi indoor positioning algorithm based on improved support vector machine [J].Computer Engineering and Applications, 2014,50(6):90-93.

[4] Nie Feng,Cheng Wei.Study on Optimal Scheduling of Multi-functional Shuttle Bus [J].Logistics Technology,2008, 27(10):251-253.

[5] Tian Jingwen, Kong Chuichao, Gao Meijuan.An improved hybrid algorithm for vehicle path planning [J]. Computer Engineering and Applications,2014,50(14):58-63.

[6] Yin Zuohai,Qiu Hong-ze,Zhou Wan-li.A Genetic Algorithm Based on Improved Mutation Operator for Solving Flexible Job Shop Scheduling [J].Computer Systems,2009,18(10): 156-159.

[7] Ji Genlin. Review of Genetic Algorithms [J]. Journal of Computer Applications and Software, 2004, 21(2):69-73.

[8] Xie Shengli, Dong Jinxiang. Solution of Job Shop Scheduling Problem Based on Genetic Algorithm [J]. Computer Engineering and Applications, 2002, 38(10): 79-82.

[9] Chen Huaping, Gu Feng, Lu Bingyuan, et al. Application of Adaptive Multi-Objective Genetic Algorithm in Flexible Job Shop Scheduling [J]. Journal of System Simulation, 2006, 18(8): 2271-2274.

[10] Long Feng.Research on the allocation and optimization of warehouse location of W company

based on adaptive genetic algorithm [D]. Guangzhou:South China University of Technology,2015.

[11] Jin Juliang, Yang Xiaohua, Ding Jing. An Improved Scheme of Standard Genetic Algorithm——Accelerated Genetic Algorithm [J]. Systems Engineering - Theory & Practice, 2001, 21(4): 8-12.

[12] Wang Tao, Fu Yili. Application of an Improved Genetic Algorithm in Job Shop Scheduling [J]. Journal of Computer Integrated Manufacturing Systems, 2002, 8(5): 392-395.

[13] Jing Bo, Liu Ying, Huang Bing. Research on JobShop scheduling problem based on genetic algorithm [J]. Journal of Computer Applications, 2013, 30(3): 688-691.

[14] Yu Qiwei. Research on Job Shop Scheduling Problem Based on Genetic Algorithm [D]. Zhejiang University, 2004.