Intelligent RGV Dynamic Scheduling Optimization Model under Possible Failure

Hong Wang¹, *, Dan Luo², Yuhe Tao¹, Tuo Feng¹ and Yuhao Xiong¹

¹Hubei University of Technology Engineering and Technology College, Wuhan 430068, China
²Wuhan University of Technology, Wuhan 430070, China

*Corresponding author email: 515971726@qq.com

Abstract. This paper analyses the dynamic RGV dynamic scheduling under the condition of possible failure. Firstly, by introducing 0-1 decision variables, the RGV position matrix and the RGV-CNC moving time distance matrix are constructed to meet the requirements of each link in the scheduling process. The time is the constraint condition, and the maximum number of processed materials and the shortest idle time of the machine tool are the objective function. The dual-objective programming model of RGV dynamic scheduling is established, and the double-objective programming problem is transformed into the single-objective programming problem by linear weighted method. For the possible fault conditions during the processing, the Poisson distribution is used to randomly generate the serial number of the CNC fault and the time of the uniform distribution to simulate the fault elimination. Then the improved genetic algorithm of the model is given, and finally the national mathematics model of 2018 is adopted. The contest B is an example and the example are verified. The test results are good.

1. Introduction
The 21st century is an information age and an intelligent era. Intelligent processing systems are widely used in various factory workshops. How to dispatch materials in the workshop to process materials, making high production efficiency and low cost is a major problem in implementing "smart workshops". The dynamic scheduling problem of RGV under the condition that the fault may occur during processing is studied.

The structure of this paper is as follows: The first part analyzes the dynamic scheduling problem of RGV in the case of CNC failure during processing; the second part establishes a dual-objective programming model for the above problem and transforms it into a single-objective planning model; it is the realization of the solution algorithm of the model. The fourth part is an example verification based on the B18 question of the national college mathematics competition in 2018.

2. Problem analysis
An intelligent processing system platform is shown in Figure 1. It consists of a number of computer numerical control machine (CNC) (8 in Figure 1, there can be more than one), and a Rail Guide Vehicle (Rail Guide Vehicle, RGV), one RGV linear track, one feeding conveyor, one feeding conveyor and
other ancillary equipment [1]. RGV is an unmanned, intelligent car that can run freely on a fixed track. It can automatically control according to the command. Move direction and distance, and bring a robot arm, two mechanical grippers and material cleaning tank, which can complete the tasks of loading and unloading and cleaning materials.

The material processing scheduling process relies on CNC and RGV to complete. At the same time, one material can only be processed on one CNC station. Each CNC has three states of processing, idle and fault. RGV can only perform movement and stop waiting at the same time. One of the loading and unloading and cleaning operations. To this end, it is necessary to introduce the CNC-processed 0-1 decision variables of the material and determine the constraints of the time required for each link in the processing, scheduling, and cleaning process. Some CNC idle time is too long (such as RGV is performing CNC2# loading and unloading operation, but CNC1# is idle). Considering improving production efficiency and reduce production cost, this paper assumes that the less idle time of CNC work, the more material production More RGV scheduling is more reasonable. Therefore, the dual objective optimization model is established with the shortest time of all CNC idle time and the maximum processing material in one operating cycle [2].

For the case where the CNC may malfunction during the processing (assuming the probability of failure is about 1%), it is assumed that each troubleshooting (manual processing, unfinished material scrapping) is between 10 and 20 minutes, troubleshooting immediately after the operation sequence is added. Due to the randomness of the fault occurrence and troubleshooting time, this paper applies the Poisson distribution to randomly generate the serial number of the CNC fault, and applies the uniform distribution to simulate the troubleshooting time.

3. Model establishment

3.1. Objective function

Set up an intelligent processing system with $N$ CNC. In order to determine the specific machining CNC of each material, introduce 0-1 decision variable $x_{ij}$, i.e.

$$x_{ij} = \begin{cases} 1, & \text{The i-th material is processed at the jth CNC} \\ 0, & \text{The i-th material is not processed in the jth CNC} \end{cases}$$

Continuous operation time of one shift The CNC has the largest number of aggregates processed within $T$ hours (excluding materials scrapped due to faults), i.e. there is an objective function
\[
\text{Max } Z_1 = \sum_{i=1}^{n} \sum_{j=1}^{N} (x_{ij} - a_{ij})
\]

(1)

Among them, \( a_{ij} = \begin{cases} 
1 & \text{The jth CNC has failed in processing the i-th material} \\
0 & \text{The jth CNC does not malfunction when processing the i-th material}
\end{cases} \)

Considering the randomness of the fault, the Poisson distribution is randomly generated here [3].

After the processing of the \( i \) th material on the \( j \) th CNC is completed, the idle waiting time of the CNC is \( t_{ij} \). Since each CNC processing time is fixed, the less idle waiting time of the CNC, the higher the efficiency of the intelligent processing system, the idle waiting time of the CNC. The sum is the smallest, that is, the following objective function

\[
\text{Min } Z_2 = \sum_{i=1}^{n} \sum_{j=1}^{N} t_{ij} \cdot x_{ij}
\]

(2)

Linearly weighting the two objective functions to transform the two-objective programming problem into a single-objective programming problem, i.e.

\[
\text{Max } Z = \omega_1 Z_1 - \omega_2 Z_2.
\]

(3)

Among them, \( \omega_1 \) and \( \omega_2 \) respectively represent the weights of linear weights, which satisfy \( \omega_1 + \omega_2 = 1 \). According to actual needs. This paper takes \( \omega_1 = 0.6 \), \( \omega_2 = 0.4 \).

3.2. Constraints

Set the \( i \)th material blanking start time to \( Tds_i \), the \( i \)th material loading start time \( Tus_i \), and the \( i \)th material to wait for idle time after processing on the \( j \)th CNC is \( t_{ij} \), the \( j \)th CNC processing material required The time is \( Tc \), the time \( Tsx_j \) required for loading and unloading, and the time constraint for the loading and unloading of the \( i \)th material is obtained by the time sequence of loading and unloading.

\[
Tus_i + t_{ij} \cdot x_{ij} + Tc \cdot x_{ij} + Tsx_j \cdot x_{ij} = Tds_i \quad (i = 1, 2, \cdots, n; \quad j = 1, 2, \cdots, N),
\]

(4)

Where \( Tds_i \) and \( Tus_i \), \( t_{ij} \) are the decision variables of the optimization model, and \( Tc \), \( Tsx_j \) is the system operation parameter [4].

Assume that \( N \) CNCs on the intelligent processing system platform are set according to the pairwise positions shown in Figure 1, and record \( y \) as the RGV position matrix.

\[
y = \begin{bmatrix} 
    y_1, y_2, \cdots, y_{\left[ \frac{N}{2} \right]}
\end{bmatrix},
\]

Among them,

\[
y_k = \begin{cases} 
1 & \text{RGV is in the kth position} \\
0 & \text{RGV is not in the kth position}
\end{cases} \quad \left( k = 1, 2, \cdots, \left[ \frac{N}{2} \right] \right)
\]
Since the position of the CNC platform where the RGV is located can only be selected at one time.

$$\sum_{k=1}^{N} y_k = 1$$

Let DRC be the RGV-CNC moving time distance matrix, whose element $DRC_{ij}$ represents the time required for RGV to move from the $i$ th CNC to the $j$ th CNC, and $T_i$ to represent the time required for RGV to move $i$ ($i=1,2,3$) units, as shown in Figure 1. There are always 8 CNCs in the intelligent system. For example, the RGV-CN C moving time distance matrix can be obtained as follows (for the case of $N$ CNCs, the corresponding moving time distance matrix can also be obtained):

$$DRC = \begin{bmatrix}
0 & T_1 & T_1 & T_2 & T_3 & T_3 \\
T_1 & 0 & 0 & T_1 & T_1 & T_2 \\
T_2 & T_1 & T_1 & 0 & 0 & T_1 \\
T_3 & T_2 & T_2 & T_1 & 0 & 0
\end{bmatrix}$$

Considering that the RGV starts feeding from the $i$ th material feeding, processing, cleaning to moving to the corresponding CNC, the time during which the time should be met is as follows:

$$Tus_{i+1} = Tus_i + Tsx_j \cdot x_{ij} + Tw + y \times DRC \times ry \quad (i = 1, 2, \cdots, n; \quad j = 1, 2, \cdots, N) \quad (5)$$

Where $Tw$ is the time required for RGV to complete a cleaning operation of a material, which belongs to the parameters given by the system. $ry$ is a matrix of $N \times 1$ consisting of $ry_j$.

$$ry = [ry_1, ry_2, \cdots, ry_N]^T \quad (6)$$

The setting of $ry_j$ is to determine whether the RGV is at the position of the $j$ CNC, and the 0-1 variable is introduced.

$$ry_j = \begin{cases}
1, \text{ RGV at the } j\text{th CNC position} \\
0, \text{ RGV is not in the } j\text{th CNC position}
\end{cases}$$

Meet the constraints

$$\sum_{k=1}^{N} ry_k = 1 \quad (7)$$

Considering the situation that the fault may occur, the fault occurrence and the troubleshooting process material loading and unloading time meet the constraints as follows

$$\left(1-a_{ij}\right) \cdot \left(Tus_i + Tsx_j + Tc + Tsx_{j+1} + T_{ij}\right) + a_{ij} \cdot \left(Ts_{ij} + T_{ij} + Tus_{i+1} + Tc + Tsx_{j+1} + Tw\right)
= \left(1-a_{ij}\right) \cdot Tds_i + a_{ij} \cdot Tds_{i+1} \quad (i = 1, 2, \cdots, n; \quad j = 1, 2, \cdots, N) \quad (8)$$
Where $T_{s_j}$ is the time when the $j$th CNC machining $i$th material failure starts, $T_{e_j}$ is the time when the $j$th CNC machining $i$th material failure ends, and $T_{g_{z_j}}$ is the $j$th CNC machining $i$th material failure when troubleshooting Time, then

$$T_{s_j} + T_{g_{z_j}} = T_{e_j} \quad (i = 1, 2, \ldots, n; j = 1, 2, \ldots, N)$$

and

$$T_{g_{z_{\min}}} \leq T_{g_{z_j}} \leq T_{g_{z_{\max}}} \quad (i = 1, 2, \ldots, n; j = 1, 2, \ldots, N) \quad (9)$$

Considering the random occurrence of the fault, the troubleshooting time is also random. Therefore, the specific CNC serial number is the determination of $a_{ij}$ when the CNC fails during the machining process, according to the number of CNCs $N$ and a certain probability (such as the failure rate). 1%, the Poisson distribution can be used to generate random numbers; the time of troubleshooting $T_{g_{z_j}}$ generally assumes that $[T_{g_{z_{\min}}}, T_{g_{z_{\max}}}]$ is uniformly distributed in a certain time interval, and this paper uniformly distributes the simulation [5].

In addition, the intelligent processing system continuously operates $T$ hours per shift, and the units involved in the above time are unified into seconds. The cutting time of all materials must not exceed $3600T$ seconds. Each material can be processed on only one CNC. The constraints are as follows,

$$\sum_{j=1}^{N} x_{ij} = 1 \quad (i = 1, 2, \ldots, n), \quad (10)$$

Cutting time constraint for the $i$th material

$$T_{u_{s_i}} < 3600T \quad (i = 1, 2, \ldots, n), \quad (11)$$

Feeding time constraint for the $i$th material

$$T_{d_{s_i}} < 3600T \quad (i = 1, 2, \ldots, n), \quad (12)$$

According to the CNC processing materials, the total quantity of clinker should be less than or equal to the maximum finished product clinker quantity within one shift of the system.

$$\sum_{i=1}^{n} x_{ij} \leq \frac{T}{T_c} \quad (j = 1, 2, \ldots, 8), \quad (13)$$

In summary, the intelligent RGV dynamic scheduling optimization model under the condition that the fault may occur is as follows.

$$\text{Max } Z_1 = \sum_{i=1}^{n} \sum_{j=1}^{N} \left(x_{ij} - a_{ij}\right)$$

$$\text{Min } Z_2 = \sum_{i=1}^{n} \sum_{j=1}^{N} t_{ij} \cdot x_{ij}$$
The above two objective functions are linearly weighted, and the bi-objective programming problem is transformed into a single-objective programming problem, that is, the following objective function is

\[
\text{Max } Z = \omega_1 Z_1 - \omega_2 Z_2.
\]

Where \( \omega_1 + \omega_2 = 1 \) is determined according to the actual situation.

The following constraints are met:

\[
\begin{align*}
Tus_i + t_{ij} \cdot x_{ij} + Tc \cdot x_{ij} + Tsx_j \cdot x_{ij} &= Tds_i \quad (i = 1, 2, \cdots n; \ j = 1, 2, \cdots, N), \\
Tus_{i+1} &= Tus_i + Tsx_j \cdot x_{ij} + Tw + y \times DRC \times ry \\ &\quad (i = 1, 2, \cdots, n; \ j = 1, 2, \cdots, N), \\
y &= \begin{bmatrix} y_1, y_2, \cdots, y_N \end{bmatrix}, \quad \sum_{k=1}^{N} y_k = 1, \\
r'y = \begin{bmatrix} r'y_1, r'y_2, \cdots, r'y_N \end{bmatrix}, \quad \sum_{k=1}^{N} r'y_k = 1, \\
\left(1 - a_{ij}\right) \cdot \left(Tus_i + TSX_j + Tc + Tsx_j + t_{ij}\right) + a_{ij} \cdot \left(Ts_{ij} + t_{ij} + Tus_{i+1} + Tc + Tsx_{j+1} + Tw\right) \\
&= \left(1 - a_{ij}\right) \cdot Tds_i + a_{ij} \cdot Tds_{i+1} \quad (i = 1, 2, \cdots n; \ j = 1, 2, \cdots, N),
\end{align*}
\]

s.t. \( Ts_{ij} + Tgz_{ij} = T e_{ij} \ (i = 1, 2, \cdots n; \ j = 1, 2, \cdots, N), \)
\( T g z_{\min} \leq Tgz_y \leq T g z_{\max} \ (i = 1, 2, \cdots n; \ j = 1, 2, \cdots, N), \)
\( \sum_{j=1}^{n} x_{ij} = 1 \quad (i = 1, 2, \cdots, n), \)
\( Tus_i < 3600T \quad (i = 1, 2, \cdots, n), \)
\( Tds_i < 3600T \quad (i = 1, 2, \cdots, n), \)
\( \sum_{j=1}^{n} x_{ij} \leq \frac{T}{Tc} \quad (j = 1, 2, \cdots, 8), \)
\( x_{ij} \in \{0, 1\}, \quad r'y_k \in \{0, 1\}, \quad y_k \in \{0, 1\}. \)

4. Improved genetic algorithm for solving scheduling optimization model
In this paper, the improved genetic algorithm is used to solve the RGV dynamic scheduling optimization model.

4.1. Coding method
In this paper, binary coding is used. According to the model constraints, each set of feasible solutions can be expressed as a material processing scheduling sequence.

4.2. Initial population generation
In genetic algorithm (GA), the initial population is generally randomly generated. In order to improve the initial population quality, the initial population should be distributed evenly in the feasible solution space [6]. According to the characteristics of the improved genetic algorithm, the generalized Hamming distance is used in the initial population. The degree of difference between individuals is quantified and satisfies \( GH_y \geq 30 \) (where \( GH_y \) is the generalized Hamming distance between individual \( i \) and \( j \)).
4.3. Design of fitness function
Since the improved genetic algorithm is based on the survival of the fittest in nature, in the iterative process, fitness is an important criterion for measuring the merits of individuals, and based on this, genetic operations are performed. Therefore, the selection of fitness function in the algorithm is used for the whole algorithm. The calculation effect has a significant impact. The fitness of the population is as large as possible. The objective function of the optimization model is the fitness function.

\[
F(x) = \alpha_1 \sum_{i=1}^{n} \sum_{j=1}^{N} (x_{ij} - a_{ij}) - \alpha_2 \sum_{i=1}^{n} \sum_{j=1}^{N} t_{ij} \cdot x_{ij}
\]

(14)

4.4. Genetic manipulation
The basic operations of improved genetic algorithms include selection, crossover, and mutation.

(1) Choice
The probability of individual fitness selection in the population is

\[
P_{i1} = \frac{F_i}{\sum_{i=1}^{N} F_i}
\]

(15)

Where \( F_i \) is the fitness value of individual \( i \) in the population and \( N \) is the population size.
The probability of difference in the individuality of the population is

\[
P_{i2} = \frac{GH_{i-queen}}{N \times GH_{avg-queen}}
\]

(16)

Where \( GH_{i-queen} \) is the generalized Hamming distance between the individual \( i \) and the queen, (indicating the difference between the individual \( i \) and the queen), and \( GH_{avg-queen} \) is the average generalized Hamming distance between the population and the queen.
The joint selection probability of individual \( i \) in the population is

\[
P_i = \alpha P_{i1} + (1 - \alpha) P_{i2}
\]

(17)

Where \( \alpha \) is the selection weight value (0 < \( \alpha \) < 1), the greater the \( \alpha \) is, the fitness is dominant, the probability of the individual with high fitness being selected is high, the convergence speed of the algorithm is faster, but it is easy to fall into local optimum; the smaller \( \alpha \) is the difference. The degree is dominant, and individuals with large differences from the queen are highly selected. Although the genetic diversity is improved, they will not fall into local optimum, but the efficiency of the algorithm is greatly reduced. The weight value should be set according to specific needs.

(2) Cross
This paper adopts a new crossover operator, which not only inherits the superior features of the parent, but also generates a child (new configuration scheme) for the scheduling problem, which is always feasible, avoiding the need for manual single point crossing. The problem of adjustment, and under the same conditions, the crossover operator is better than other crossover operators based on process coding [7].

(3) Variation
Reverse order mutation is used, that is, a small segment of the gene is randomly selected, and then the gene is arranged in reverse order while other genes remain unchanged.
4.5. **Introducing the catastrophe operator**  
In this paper, we choose to increase the mutation rate and set a catastrophe countdown $T$ at the beginning of the algorithm. During an iterative process, the gene of the queen bee changes, indicating that the algorithm has strong search ability. If the gene of the queen king does not change, then it shows that the algorithm may fall into the local optimal solution, and the catastrophe timer is decremented by 1. When the timer returns to zero, the probability of catastrophe increasing the mutation occurs, and a new solution space is developed for the algorithm. If the gene of the queen bee new before the catastrophe occurs the change indicates that the solution space of the current search still has the value of exploration, and the timer is reset for the next iteration.

5. **Instance verification**  
In order to verify the practicability of the RGV dynamic scheduling optimization model and the effectiveness of the algorithm, the following is a third question of the National College Student Digital Model Competition B18 in 2018. The scheduling model and algorithm established in this paper are used to solve the CNC under one process. Material processing operations in the event of a fault.

In the 2018 National College Student Digital Model Competition B, the intelligent system (see Figure 1) has a total of 8 CNCs, which may be faulty during processing (according to statistics: the probability of failure is about 1%), each time Troubleshooting (manual processing, unfinished material scrapping) time is between 10~20 minutes, and the operation sequence is added immediately after troubleshooting [8]. Table 1 gives three sets of data of intelligent processing system operating parameters.

**Table 1.** 3 sets of data tables for intelligent processing system operating parameters

| System operation parameters | 1  | 2  | 3  |
|-----------------------------|--|--|--|
| RGV takes 1 unit to move time | 20 | 23 | 18 |
| RGV takes 2 unit to move time | 33 | 41 | 32 |
| RGV takes 3 unit to move time | 46 | 59 | 46 |
| The time required for CNC to process materials for one process | 560 | 580 | 545 |
| The time required for CNC to complete the first process of a two-process material | 400 | 280 | 455 |
| The time required for CNC to complete the second process of a two-process material | 378 | 500 | 182 |
| RGV is the time required for CNC1#, 3#, 5#, 7# to load and unload at one time | 28 | 30 | 27 |
| RGV is the time required for CNC2#, 4#, 6#, 8# to load and unload at one time | 31 | 35 | 32 |
| RGV takes time to complete a material cleaning operation | 25 | 30 | 25 |

Note: 8 hours of continuous operation per shift.

For the three sets of data given in Table 1, the above scheduling model and algorithm are applied respectively, and the scheduling strategy of RGV is obtained through C# language programming (limited to space, here), and the maximum number of processed materials is 326, 327, 312, respectively. The operating efficiency is 92.61%, 92.89%, and 88.63%, respectively, which illustrates the effectiveness of the RGV dynamic scheduling model and algorithm in the case of CNC failure.

**Acknowledgments**  
Wang Hong was born in June 1972, Wuhan, female, Han nationality. Her degree is a master's degree and she is an associate professor. She graduated from Wuhan University of Technology and graduated with applied mathematics. She is major in mathematical statistics/big data. She currently works at Hubei University of Technology Engineering and Technology College.

Luo Dan is Master and lecturer who works in School of Mechanical and Electronic Engineering, Wuhan University of Technology.

Tao Yuhe, Feng Tuo, Xiong Yuhao are undergraduate students of School of Mechanical and Electronic Engineering, Wuhan University of Technology.
References

[1] Wu Jianjun, Nie Guodong. On the Development of College Students' Integrity Education under the Guidance of Socialist Core Values. Educational Exploration, Vol. 4 (2015) No.28, p. 103 - 105.

[2] Si Shoukui, Sun Zhaoliang, Mathematical Modeling Algorithm and Application. National Defense University Press, 2015.

[3] Zhao Dongfang. Mathematical Models and Competitions. Huazhong Normal University Press, 2014.

[4] Cheng Guohuang, Xia Shengxiong, Wang Yu, et al. Workshop real-time scheduling system based on multi-group hybrid genetic algorithm. Ship Science and Technology, Vol. 11 (2011) No.28, p. 126 - 130.

[5] McKay K N, Wiers V C S. Unifying the Theory and Practice of Production Scheduling. Journal of Manufacturing System, Vol. 4 (1999) No.18, p. 241 - 255.

[6] Gu Wen Bin, Tang Dun Bing, Zun Kun. Minimizing Makespan in Job-shop Scheduling Problem Using an Improved Adaptive Particle Swarm Optimization Algorithm. Chinese Control and Decision Conference (CCDC): Tai Yuan, 2012.

[7] Wang Ling. Workshop Scheduling and Its Genetic Algorithm. Beijing: Tsinghua University Press, 2003.

[8] Zhang Chaoyong, Rao Yunqing et al. Genetic algorithm based on POX crossover to solve Job-Shop scheduling problem. China Mechanical Engineering, Vol. 1 (2004) No.23, p. 149 - 153.