


Fitness Distance Correlation Strategy for Solving the RGV Dynamic Scheduling Problem

Wei Li, Jiangxi University of Science and Technology, China

 <https://orcid.org/0000-0001-8242-076X>

Furong Tian, Jiangxi University of Science and Technology, China

Ke Li, Jiangxi University of Science and Technology, China

ABSTRACT

Rail guide vehicle (RGV) problems have the characteristics of fast running, stable performance, and high automation. RGV dynamic scheduling has a great impact on the working efficiency of an entire automated warehouse. However, the relative intelligent optimization research of different workshop components for RGV dynamic scheduling problems are insufficient scheduling in the previous works. They appear idle when waiting, resulting in reduced operating efficiency during operation. This article proposes a new distance landscape strategy for the RGV dynamic scheduling problems. In order to solve the RGV dynamic scheduling problem more effectively, experiments are conducted based on the type of computer numerical controller (CNC) with two different procedures programming model in solving the RGV dynamic scheduling problems. The experiment results reveal that this new distance landscape strategy can provide promising results and solves the considered RGV dynamic scheduling problem effectively.

KEYWORDS

Computer Numerical Controller, Dynamic Scheduling, Fitness Distance Correlation, Fitness Landscape, Rail Guide Vehicle

1. INTRODUCTION

With the development of control engineering technology, many enterprises have gradually raised the awareness of industrial intelligent automation. Rail guide vehicle dynamic scheduling problem is an important branch in the automation industry. Nowadays, the RGV is mainly applied in production scheduling workshop, logistics transportation, component assembly, and many other fields. The RGV has the characteristics of fast running, stable performance and high automation, the dynamic scheduling problem associated with RGV can effectively improve the production efficiency of modern intelligent processing, which has been widely used in various workshops and automated warehouses. The RGV can be divided into self-driven type, passive-driven type, assembly type and transport type according to the driving mode and the purpose (Martina et al., 2018; Sáez et al., 2008). At present,

DOI: 10.4018/IJCINI.2020070102

This article, originally published under IGI Global's copyright on July 1, 2020 will proceed with publication as an Open Access article starting on February 1, 2021 in the gold Open Access journal, International Journal of Cognitive Informatics and Natural Intelligence (converted to gold Open Access January 1, 2021), and will be distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

most enterprises choose the ring RGV, which can run multiple vehicles on the same track and greatly improve the capacity of transportation and assembly (Lee et al., 1996).

The research on the RGV dynamic scheduling is mainly based on the warehousing system (Roy et al., 2016). Lee et al. proposed the RGV scheduling strategy based on FCFS and analyzed the system operating efficiency of different quantities the RGV in automated warehousing system (Zhu et al., 2016). Sáez et al. (2008) effectively completed the multiple RGV scheduling tasks by predicting the tasks to be generated in advance, fuzzy classification algorithm was used to calculate and generate the probability of the tasks based on historical data, and then used a genetic algorithm to find a reasonable RGV scheduling path (Gagliardi et al., 2015). Zhang et al. established the genetic algorithm model and made a comprehensive analysis of the dynamic scheduling model of loading and unloading completed by the cooperation between the RGV and CNC (Ferrara et al., 2014). Wu et al. constructed a scheduling model with the shortest running time of RGV as the objective function, and then they adopted dynamic programming algorithm based on TSP to save the scheduling time of the RGV, accordingly improved the production efficiency of intelligent machining (Roy et al., 2016). All the above researches were carried out purely from the perspective of traditional algorithms, which is used to solve the scheduling problem by setting various constraints. Meanwhile, traditional algorithms are Although the scheduling results sometimes have some rationality using traditional algorithms, they are easy to fall into the local optimum, higher complexity, and more runtime.

The fitness landscape is a theory applied to the optimization dynamics of biological evolution proposed by Wright (Wright, 1932). Study of the fitness landscape is an important topic of evolutionary computation. Influenced by biological evolution, researchers began the fitness landscape research early in the field of evolutionary computation, whose purpose is to understand the behavior of evolutionary algorithms to solve optimization problems. The fitness landscape can reveal the relationship between the search solution space and fitness by the features of the landscape information, which regards evolutionary optimization as a process of an adaptive random walk on the three-dimensional landscape visualized by ridges, canyons, and basins. If the search space is regarded as a landscape, the evolutionary algorithm can be understood as navigation through the landscape to locate its highest peak. The fitness landscape is described as mapping from a set of genotypes, and it can be considered as the height of the entire genotype. In other words, the fitness involved in the fitness landscape is an orthogonal projection that can be expressed as a genotype attribute. However, the search strategy of the evolutionary algorithm is to decode the genotype into a phenotype with high fitness. The fitness landscape can be represented as a common metaphor used to describe the behavior of evolutionary algorithms in the solution of optimization problems. The fitness landscape is a visual tool with geometric meaning that can be used to evaluate population characteristics as they change over time. These changes are achieved by mapping one genotype to another. Many studies and discussions of the fitness landscape topology have been published, including one in which the fitness landscape topologies are related to the optimization problem of evolutionary computation. The problem-oriented fitness landscape has been deeply analyzed, especially for combinatorial optimization problems, such as the Traveling Salesman Problem (Angel & Zizzimopoulos, 2001), Image Segmentation (Bouziri et al., 2011), the Graph Coloring Problem (Daolio et al., 2010), the Quadratic Assignment Problem (He & Yao 2005; Malan & Engelbrecht 2009), the MAX-SA (Merz & Freisleben, 2000), and the Knapsack Problem (Prugel-Bennett & Tayarani-Najaran, 2012). Although the main focus of these studies is to provide the most effective information to solve problems in evolutionary computation, the experimental results of some optimization problems also show the features of various fitness landscape topologies. For example, in the symmetric TSP problem, the travel cost from i point to j point is the same as j to i , and the corresponding fitness landscape topology is quite smooth. In comparison, the fitness landscape presented by the asymmetric TSP problem is quite rough (Reidys & Stadler, 2001). For the quadratic assignment problem, the experiment shows that a significant neutrality degree of fitness landscape topography can be obtained (Shen & He, 2010). For the continuous real-valued

problem, similar numerical results can be obtained with fitness landscape topography among widely used benchmarks (Caamaño et al., 2013).

With the continuous improvement in the optimization solution of the evolutionary algorithm, the fitness landscape can present more abundant feature information around fitness, which is mainly shown in the local fitness, fitness distance correlation, random walk, and landscape roughness. These fitness landscape features reflect the optimal solution distribution, quantity, and local unimodal or multimodal topological structures from a different perspective. Therefore, this work analyzes the relevant fitness landscape features and then establishes the RGV dynamic scheduling model. In order to solve the RGV dynamic scheduling problem more effectively, this paper using a distance landscape strategy by combining the fitness landscape and dynamic search strategy. Experiments are conducted based on the type of CNC with two procedures programming model in solving the RGV dynamic scheduling problems. The experiment results reveal that this new distance landscape strategy can provide promising results and solve the considered RGV dynamic scheduling problems effectively.

The remainder of the paper is organized as follows. Section 2 briefly introduces fitness landscape analysis and fitness distance correlation. Section 3 illustrates the model description of RGV dynamic scheduling problem. Section 4 presents the design and implementation of the proposed distance landscape strategy for the RGV dynamic scheduling problem. The detail of experimental analysis and verification are described in section 5. The conclusions are summarized in the final section include some topics of the future research are provided.

2. FITNESS LANDSCAPE ANALYSIS

2.1. Definition of the Fitness Landscape

Study of the fitness landscape is an important area of evolutionary computation, whose purpose is to explain the behavior of evolutionary algorithms in the solution of optimization problems. Influenced by biological evolution, researchers began to study the fitness landscape early in the field of evolutionary computation. A static fitness landscape can be represented in an abstract manner with triples (Kallal et al., 2000) as follows:

$$L = (X, f, N_k) \quad (1)$$

where the tuple X is a feasible solution space set of the evolutionary algorithm. The fitness function $f: X \rightarrow R$ is used to calculate the fitness value of the priority observation $x \in X$ in the fitness landscape. The neighborhood tuple N_k given by the distance metric of size k is defined on the tuple set X , and N_k has $N_k(x) = \{y \in X : d(x, y) \leq k\}$.

2.2. The Topology of the Fitness Landscape

The fitness landscape topological structures can be enlightening and instructive for the population structure of the evolutionary algorithm, the recombination mutation strategy of the search space, and even algorithm performance optimization. However, the topological structures also clearly show the limitation of geometric conception, especially, the above fitness landscape topologies are based on two dimensions. If the genotype space is higher than two dimensions, the geometric topological features of the fitness landscape are difficult to represent. In fact, the fitness landscape topological features exist for all dimensions, only with the dimension increases, the meaning of the feature may change, and the geometric feature will present more complex and variable pattern. This situation can be converted into a two-dimensional landscape topology by using the methods that include dimensionality reduction and mapping operation. The local fitness landscape topology looks like ridges, canyons, plateaus, and basins, there are directly viewed as having the same meaning as two-dimensional landscapes.

Therefore, through discussing the problem which whether two-dimensional landscape features such as fitness distance correlation, fitness information roughness, and gradient neutrality are related to higher dimensions have great important research value for evolutionary algorithm optimization. In order to better solve the representation of the fitness landscape, this work focus on the use of fitness distance correlation representation to analyze characteristics of the optimization problem.

2.3. Fitness Distance Correlation

The genotype of the fitness landscape is reflected from one variant to another in the distance attribute. An important part of solving this distance attribute problem is to collect possible topological and structural features in the fitness landscape and to analyze and express them into the potential and observable mechanisms in evolutionary algorithms. The fitness distance correlation (FDC) analysis method can quantify the distance relationship between the fitness of a set of points in the local fitness landscape and the global minimum (Müller & Sbalzarini 2011). This section will use the fitness distance correlation analysis method to represent the topological features of the fitness landscape. The customized triple of the fitness landscape F is defined as:

$$F = (\chi, d_E, f) \quad (2)$$

where the local effective fitness landscape domain χ is constrained by $\chi = [\iota, u] \subset R^n$, vector ι , $u \in R^n$ defines the upper and lower bounds. The distance d_E between points are calculated by using the Euclidean distance in the fitness landscape decision space. From the local fitness landscape to evaluate the fitness function f in the statistical sampling region given a random sample $x(j) \in \chi$ and $f(j) \in R^n$, $j = 1, \dots, s$. In the standard fitness distance correlation analysis, the global minimum x_{\min} is considered to be a priori known, and x_{\min} is usually approximated by $\tilde{x}_{\min} = \arg \min [x(j)f(x(j))]$, $j = 1, 2, \dots, s$. The features presented by the distance $d^{(j)} = d_E(x_{\min}, x^{(j)})$ or $d^{(j)} = d_E(\tilde{x}_{\min}, x^{(j)})$ and the corresponding fitness value $f^{(j)}$ can provide a speculative means for the global topology of the fitness landscape.

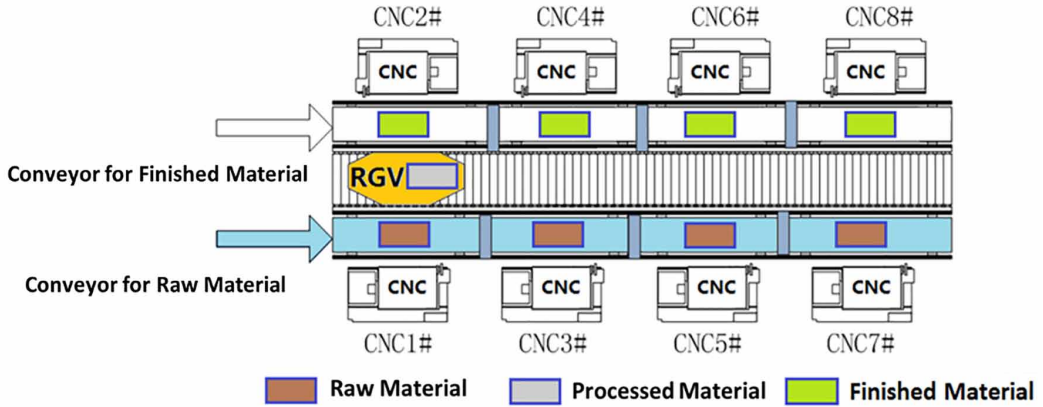
3. THE RGV DYNAMIC SCHEDULING PROBLEM DESCRIPTION

In the following section, in order to discuss and analyze the RGV dynamic scheduling problem more conveniently, we construct an example simplified simulation scenario. There are two kinds of materials that can be processed by the intelligent RGV scheduling system, one process can be completed in single-procedure scheduling, and the other process requires multi-procedure scheduling to complete. This scheduling system consists of 8 CNCs, 1 RGV, 1 RGV linear track, 1 loading conveyor belt, 1 unloading conveyor belt, and other accessorial components. A sample of the RGV dynamic scheduling system is described by Figure 1.

It is assumed that this type of the RGV can obtain the real-time working status of each CNC by scheduling the historical demand signal, comprehensively consider and compare the remaining scheduling time of each CNC, the loading and unloading time, and the required time for the RGV to the corresponding CNC position, the task order is under determined. This type of the RGV start in the next round of optimal CNC, all CNCs demand to wait within 8 hours.

For the convenience of describing the RGV dynamic scheduling problem, some necessary assumptions are given as follows:

Figure 1. A sample of the RGV dynamic scheduling system



- In view of the safety requirements, the tools cannot be replaced midway until the system stops working after the scheduling system is started;
- When the RGV is operated to a certain CNC where work is required, the loading conveyor will deliver the raw material to the front of the CNC at the same time, and the unloading conveyor will immediately send the cleaned material;
- All RGV calculation processing times are considered to be zero, which does not have any effect on system performance;
- Unless the scheduling system stops working for 8 hours of continuous operation, the operation of RGV such as moving, loading, unloading and cleaning will not be stopped midway.

The symbols that need to be used in the modeling process are unified in Table 1.

3.1. RGV Single-Procedure Scheduling Model

In the intelligent processing system, one RGV is dispatched to carry out loading and unloading and washing of 8 CNC machines arranged on both sides. The loading and unloading of the n th round only have two status of going forward and stopping, we introduce 0 and 1 variables expression as follows:

$$f_move_{i,n} = \begin{cases} 1, & \text{the } n\text{th round of loading and unloading goes to } CNCi\# \\ 0, & \text{the } n\text{th round of loading and unloading doesn't go to } CNCi\# \end{cases} \quad (3)$$

Comparing the remaining processing time of each CNC machine before each round of movement $t_rest_{i,n}$ with the time required before moving to each CNC machine t_move_n , the larger value plus loading and unloading time $t_add_{i,n}$ is the time taken by the RGV from heading to $CNCi\#$ complete the n th round of loading and unloading on $CNCi\#$:

$$t_{i,n} = \max\{t_rest_{i,n}, t_move_n\} + t_add_{i,n} \quad (4)$$

Before the RGV starts working, all CNCs are idle, which is different from the status after working, so these different statuses will be analyzed separately. In order to distinguish between the first 8

Table 1. The symbol description of RGV dynamic scheduling model

Symbol	Symbol Description
$nth, n = 1, 2, \dots$	The serial number of loading and unloading rounds.
$CNCi \#, i = 1, 2, \dots, 8$	Refers to $CNC1 \#, CNC2 \#, \dots, CNC8 \#$
$f_move_{i,n}$	$f_move_{i,n}$ determines whether to go to the $CNCi \#$ workbench before the nth round of loading and unloading status.
$f_have_{i,n}$	$f_have_{i,n}$ determines whether there any materials on the $CNCi \#$ workbench before the nth round of loading and unloading status.
s, e	$s, e = 1, 2, 3, 4$ denotes the starting and ending position of the RGV movement, respectively. When $s, e = 1$, the position of the RGV is between $CNC1 \#$ and $CNC2 \#$. When $s, e = 2$, the position of the RGV is between $CNC3 \#$ and $CNC4 \#$. When $s, e = 3$, the position of the RGV is between $CNC5 \#$ and $CNC6 \#$. When $s, e = 4$, the position of the RGV is between $CNC7 \#$ and $CNC8 \#$. s_n and e_n respectively denote the start and end positions of the RGV when nth round is loaded.
$t_{i,n}$	The nth round took time of the RGV for reaching $CNCi \#$ from loading and unloading position.
t_move_n	The nth round took time from s_n to e_n .
$t_add_{i,n}$	The nth round took time of the RGV for loading and unloading at the $CNCi \#$ position.
$t_wash_{i,n}$	The nth round took time of the RGV for cleaning at the $CNCi \#$ position.
$t_rest_{i,n}$	The nth round took time of the RGV for processing the remaining materials at the $CNCi \#$ position.
A_n	After the nth round of loading and unloading, the number of raw materials produced in one process.
AB_n	After the nth round of loading and unloading, the number of raw materials produced in two processes.
$order_i$	The $CNCi \#$ responsible for the i -th processing process, $i = 0, 1, \dots, n$.
$left_n$	The number of materials for the target process in the nth round of loading and unloading.

rounds of working after no washing stage, set $f_have_{i,n}$ to determine whether $CNCi\#$ has material before loading and unloading of the n th round on the working table:

$$f_have_{i,n} = \begin{cases} 1, & \text{there are materials on } CNCi\# \text{ work table} \\ 0, & \text{there aren't materials on } CNCi\# \text{ work table} \end{cases}, n \leq 8 \quad (5)$$

The material needs to be cleaned after each unloading, using $t_wash_{i,n}$ denote the cleaning time of the n th round.

The single-procedure programming model with the largest quantity of finished materials can be established as follows:

- The decision variable:

$$f_move_{i,n} \quad (6)$$

- The objective function:

$$\max A_n \quad (7)$$

A_n : the total amount of finished materials obtained after loading, processing, washing, and unloading in n rounds.

The constraint conditions:

- The RGV moves to the CNC with the shortest waiting time each round by the following problem:

$$\max \{f_move_{i,n} \cdot t_{i,n}\} = \min t_{i,n} \quad (8)$$

- One RGV can only load and unload and wash one material at a time, we have:

$$\sum_{i=1}^8 f_move_{i,n} = 1 \quad (9)$$

- Once one material is acquired by CNC, the material on the CNC working table will always exists, then we have:

$$f_have_{i,n} = f_have_{i,n} + f_move_{i,n} (1 - f_have_{i,n}) \quad (10)$$

- Each shift is operated continuously for 8 hours, the total working time of the CNC and RGV should not exceed 8 hours. This implies that:

$$\sum_{n=1}^n \sum_{i=1}^8 f_move_{i,n} \cdot (t_{i,n} + t_wash_{i,n} \cdot f_have_{i,n}) \leq 8 \times 3600 \quad (11)$$

- The starting position of s_{n+1} is the ending position of e_n in the n th round, then we have:

$$s_{n+1} = e_n \quad (12)$$

- After each round of loading and unloading, the cumulative output of finished materials can be increased by up to 1, we have:

$$A_{n+1} = A_n + \sum_{i=1}^8 (f_have_{i,n} \cdot f_move_{i,n}) \quad (13)$$

- Set the initial value, when the first round of loading and unloading, the quantity of finished materials is 0:

$$A_1 = 0 \quad (14)$$

3.2. RGV Multi-Procedure Scheduling Model

For materials with two processing procedures, there are three states in the machining process: raw material, processed raw material and finished material. In a shift, when a CNC can only use one tool to process one procedure, how to make RGV identify the state of the material held by the mechanical arm when loading and unloading, and the next step to deal with it is a big difficulty. To solve these two problems, the following two parameters are introduced: $order_i$ refers to the processing type of $CNCi \#$, $left_i$ refers to the states of material in RGV. Only when these two parameters can be successfully matched can RGV proceed to the next step, the loading and unloading of the second procedure.

As for the CNC's processing tool distribution, because there must be two procedures. There are $(C_8^1 + C_8^2 + \dots + C_8^7) = 254$ ways to allocate the tools. Different distribution schemes will have a great impact on the working time and quantity of finished material of RGV. Therefore, this paper established a mutli-procedure programming model with RGV scheduling path as the decision variable to obtain the maximum quantity of finished materials are as follows:

- The decision variable:

$$f_move_{i,n} \quad (15)$$

- The mutli-procedure functions:

$$\max AB_n \quad (16)$$

$$\min \sum_{n=1}^n \sum_{i=1}^8 f_move_{i,n} (t_{i,n} + t_wash_{i,n} \cdot f_have_{i,n}) \quad (17)$$

AB_n : the quantity of finished materials after completing two procedures.

The constraint conditions:

- In one processing shift, a CNC can only install one tool and process one procedure, and it is not allowed to change the tool in the middle of the process for safety, namely:

$$order_i = \begin{cases} 0, & \text{responsible for the first procedure} \\ 1, & \text{responsible for the second procedure} \end{cases} \quad (18)$$

- There are only two states of material in RGV during the working period, namely:

$$left_n = \begin{cases} 0, & \text{no material in RGV} \\ 1, & \text{having a cleaned material in RGV} \end{cases} \quad (19)$$

- The n th round of loading and unloading can only be carried out when the material is successfully matched with the $CNCi \#$ process ($order_i = left_n$), namely:

$$f_move_{i,n} = f_move_{i,n} (1 - |order_i - left_n|) \quad (20)$$

- If N_1 is the number of CNC responsible for the first procedure and N_2 is the number of CNC responsible for the second procedure, then $N_1 + N_2 = 8, N_1, N_2 \in N^*$.

4. DISTANCE LANDSCAPE STRATEGY FOR THE RGV DYNAMIC SCHEDULING PROBLEM

The distance landscape strategy will be discussed in this section. According to the fitness distance correlation method, the local fitness landscape features will be represented and analyzed for the dynamic scheduling problem. The distance landscape strategy relationship of the RGV dynamic schedule problem will be designed and performed.

4.1. Distance Landscape Strategy

For a population $P = \{\bar{X}_1, \bar{X}_2, \dots, \bar{X}_\omega\}$, where each $\bar{X}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,n})$ is a solution on R^n . According to the FDC method and the correlation coefficient, the detailed steps of distance landscape strategy are as follows:

Step 1: Finding the optimal solution in the current population and denoting it as \bar{X}^* , then calculating the distance between each $\bar{X}_i, i = 1, 2, \dots, \omega$, and the optimal solution \bar{X}^* by the following formula:

$$d(\bar{x}_i, \bar{x}^*) = \sum_{j=1}^n |x_{i,j} - x_j^*| \quad (21)$$

Step 2: The set of $\{\bar{X}_1, \bar{X}_2, \dots, \bar{X}_\omega\}$ corresponding to the distance value $d(\bar{x}_i, \bar{x}^*)$ calculated above is sorted according to the rule from low to high, and it is denoted as $k_1, k_2, \dots, k_\omega$ in order.

Step 3: Initialization: Set $\theta = 0$ to calculate the fitness function $f_{k_i}, i = 1, 2, \dots, \omega$, if $f_{k_{i+1}} + \varepsilon \leq f_{k_i}$, then $\theta = \theta + 1$, where ε is the error value. This value is added to maintain the diversity of the population and avoided the local optimum or global convergence in the late stage of optimization.

Step 4: Normalizing θ :

$$\frac{\theta}{\mu - 1} \quad (22)$$

where μ is the population size in the local fitness landscape, $0 \leq \varphi \leq 1$. According to the analysis and representation of landscape features: When ϕ close to 0, the local fitness landscape is closer to the uni-modal topology, and when ϕ close to 1, the local fitness landscape is closer to the multimodal topology.

4.2. Process of Solving the RGV Scheduling Model

It is difficult to obtain the global optimal solution directly through the model, we design a heuristic algorithm to obtain the approximate optimal solution by the distance landscape strategy for the shortest loading and unloading of each round. The basic steps and principles of the algorithm are described in Algorithm 1.

Algorithm 1: Process of solving the single-procedure RGV scheduling model

input:

- The cumulative time: **T**;
- The initial number of finished materials: **Count**;
- Some known running and operation time.

output:

- The number of finished materials: **Count**;

while **T** < 8 hours **do**

Calculate the moving and operation time from the current position of RGV to 8 CNCs; the RGV go to the CNC that takes the least time;

if this CNC has material **do**

RGV completes loading and unloading, start cleaning;

Count ++;

else

RGV completes loading;

end

Update the position of RGV, the remaining working time of each CNC, **T** plus the time consumed by the RGV in this cycle;

end

From the beginning of processing for solving the single-procedure RGV scheduling system, by considering all historical demand signals and current demand signals, we designed this algorithm to find the CNC that completes the loading and unloading, and respond to the current demand signal of the CNC, so that the CNC completes the process of loading and unloading. Then the algorithm judges whether to wash the material according to the actual situation. Repeat the above operation. When the accumulated time exceeds 8 hours, the iteration is terminated.

For the processing of the two procedures, each CNC can only install one tool to process one material at the same time, and the tool cannot be replaced during the processing period. The first and second procedures need to be processed successively on different CNC, and the completion time is also different. Each CNC can only complete one procedure. Therefore, it is of great importance for the reasonable distribution of processing tools.

For each tool distribution scheme, the heuristic algorithm is constructed with the same idea as the algorithm under one procedure to obtain the corresponding approximate optimal solution. According to the multi-procedure, selecting the optimal tool distribution scheme and the corresponding material processing situation. Every distribution scheme must exist tool for procedure one and tool for procedure two simultaneously, not allowing all CNCs to install the same tool in a shift. Then traverse 254 distribution schemes and solve them according to Algorithm 2.

Algorithm 2: Process of solving the multi-procedure RGV scheduling model

input:

- The cumulative time: T ;
- The initial number of finished materials in each case: $Count_i (i = 1, 2, \dots, 254)$;
- Some known running and operation time;
- The number of all possible tool distributions: N ;
- CNC's current target procedure: **flag**;

output:

- The number of finished materials in each case: $Count_i$;

for $i \leftarrow 1$ **to** N **do**

while $T < 8 \text{ hours}$ **do**

calculate the moving and operation time from the current position of RGV to 8 CNCs;

RGV go to the CNC that takes the least time;

if *this CNC responsible for procedure 1* (**flag**=0) **do**

if *this CNC have workpiece* **do**

RGV completes the loading and unloading operation;

flag =1;

else

RGV completes the loading operation;

end

else

if *this CNC have workpiece* **do**

RGV completes the loading and unloading operation, start cleaning;

$Count_i ++$;

flag =0;

else

RGV completes the loading operation;

flag=0;

end

end

update the position of RGV, the remaining working time of each CNC, T plus the time consumed by RGV in this cycle;

end

end

From the beginning of processing for solving the multi-procedure RGV scheduling system, by considering all historical demand signals and current demand signals, finding out the fastest CNC to complete the loading and unloading among eight CNC whose current target procedure is accord with the procedure it is responsible for RGV should respond to the CNC when demand signal comes, heading to this CNC to complete the loading and unloading process, then determine whether to wash the material in accordance with the actual situation and whether to change the target procedure. Repeat the above procedure until the accumulations exceed 8 hours, the loop is stopped. According to the double decision objective, the Algorithm 2 can find out the optimal tool distribution scheme and the corresponding material processing condition.

5. EXPERIMENTAL RESULTS AND ANALYSIS

In order to test the effectiveness of the RGV dynamic scheduling model based on the distance landscape strategy, we use three different sets of system operating parameters to calculate and analyze the results to evaluate the operational efficiency of the RGV scheduling strategy.

5.1. Data Initialization

According to the characteristics of RGV in real-life scenarios, combined with the actual requirements of the RGV dynamic scheduling system, the three sets of parameters for initializing the intelligent machining system are shown in the Table 2. The data in this table reflects the time required for the RGV and CNC work processes.

Table 2. Parameter settings of system operation

System Operation Parameters	Group 1	Group 2	Group 3
RGV to move 1 unit	20	23	18
RGV to move 2 units	33	41	32
RGV to move 3 units	46	59	46
CNC to complete signal process	560	580	545
CNC1#, 3#, 5# and 7# loading and unloading	28	30	27
CNC2#, 4#, 6# and 8# loading and unloading	31	35	32
RGV to complete a material cleaning operation	25	30	25

5.2. The Experimental Results

According to the provided system parameters and three sets of data, combined with the above algorithm process, the scheduling results obtained by MATLAB software from Tables 3 to 8.

From Table 3 to Table 5 the results show that the quantity and specific execution time of materials can be processed by the scheduling system during processing a procedure within 8 hours from the beginning to the end. From these tables, we can find that in the same processing time, the number of processing materials under the third group of parameters is the largest, reaching 392; the second group is the least, only 359.

From Table 6 to Table 8, the results show that quantity and specific execution time of materials can be processed by the scheduling system during processing two procedures within 8 hours from the beginning to the end. Because there are two processing procedures, the quantity of processed materials is obviously less than that of one procedure. Despite this, the quantity of processing materials

Table 3. The results of the first group under one procedure

The Number of Materials	The Number of CNCs	The Time of Loading	The Time of Unloading
1	1	0	588
2	2	28	641
3	3	79	717
4	4	107	770
5	5	158	846
6	6	186	899
7	7	237	975
8	8	265	1028
9	1	588	1176
10	2	641	1232
⋮	⋮	⋮	⋮
380	4	27956	28547
381	5	28032	28623
382	6	28085	28676

Table 4. The results of the second group under one procedure

The Number of Materials	The Number of CNCs	The Time of Loading	The Time of Unloading
1	1	0	610
2	2	30	670
3	3	88	758
4	4	118	818
5	5	176	906
6	6	206	966
7	7	264	1054
8	8	294	1114
9	1	610	1238
10	2	670	1446
⋮	⋮	⋮	⋮
357	5	27910	28538
358	6	27970	28598
359	7	28058	28686

Table 5. The results of the third group under one procedure

The Number of Materials	The Number of CNCs	The Time of Loading	The Time of Unloading
1	1	0	572
2	2	27	624
3	3	77	699
4	4	104	751
5	5	154	826
6	6	181	878
7	7	231	953
8	8	258	1005
9	1	572	1114
10	2	624	1201
⋮	⋮	⋮	⋮
390	6	27997	28574
391	7	28072	28649
392	8	28124	28701

Table 6. The results of the first group under two procedures

The Number of Materials	The Number of CNCs With Procedure One	The Time of Loading	The Time of Unloading	The Number of CNCs With Procedure Two	The Time of Loading	The Time of Unloading
1	1	0	428	2	456	898
2	3	48	532	4	560	1002
3	5	96	636	6	664	1106
4	7	144	740	8	768	1210
5	1	428	870	2	898	1340
6	3	532	974	4	1002	1444
7	5	636	1078	6	1106	1548
8	7	740	1182	8	1210	1652
9	1	870	1312	2	1340	1782
10	3	974	1416	4	1444	1886
⋮	⋮	⋮	⋮	⋮	⋮	⋮
251	5	27598	28040	6	28068	28510
252	7	27702	28144	8	28172	28614
253	1	27832	28274	2	28302	28744

Table 7. The results of the second group two procedures

The Number of Materials	The Number of Cncs With Procedure One	The Time of Loading	The Time of Unloading	The Number of CNCs With Procedure Two	The Time of Loading	The Time of Unloading
1	2	0	315	1	350	880
2	4	58	433	3	468	998
3	6	116	551	5	586	1116
4	8	174	669	7	704	1275
5	2	315	940	1	880	1429
6	4	433	805	3	998	1574
7	6	551	1081	5	1116	1665
8	8	669	1335	7	1275	1805
9	2	805	1199	1	1429	1959
10	4	940	1489	3	1537	2077
⋮	⋮	⋮	⋮	⋮	⋮	⋮
209	8	27305	27835	1	27929	28459
210	2	27459	27989	3	28047	28577
211	4	27577	28107	5	28165	28695

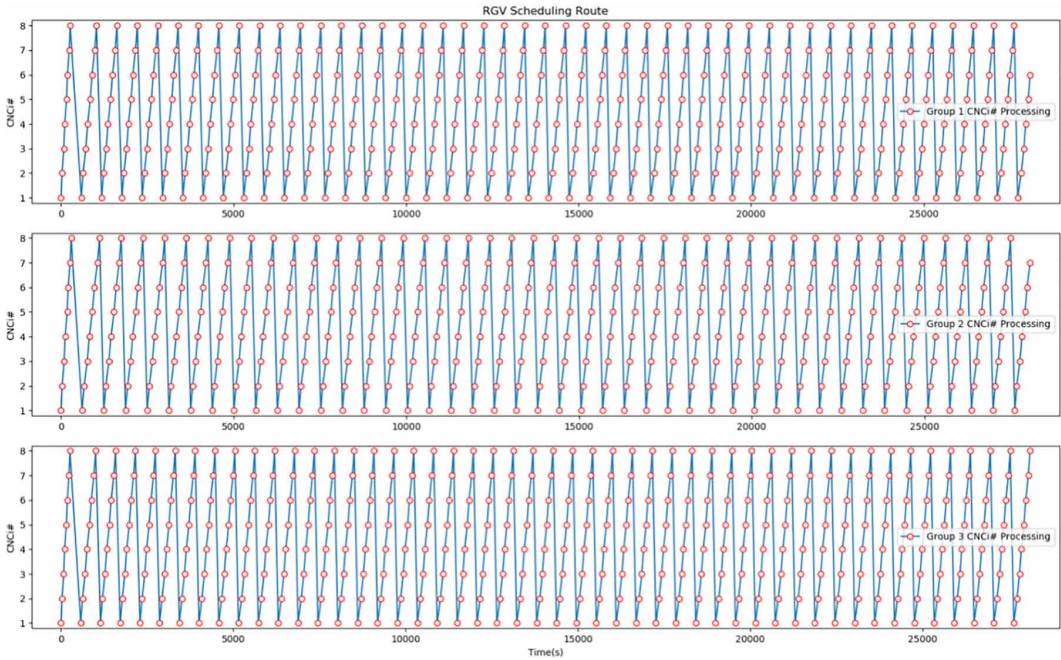
Table 8. The results of the third group under two procedures

The Number of Materials	The Number of CNCs With Procedure One	The Time of Loading	The Time of Unloading	The Number of CNCs With Procedure Two	The Time of Loading	The Time of Unloading
1	1	0	482	2	509	958
2	3	45	584	5	629	838
3	4	72	908	2	958	1189
4	6	122	681	8	731	1407
5	7	172	788	5	838	1074
6	1	58	1015	5	1074	1310
7	3	482	1144	2	1189	1671
8	6	584	1278	5	1310	1546
9	7	681	1380	8	1407	1889
10	4	788	1496	5	1546	1787
⋮	⋮	⋮	⋮	⋮	⋮	⋮
242	1	27576	28132	5	28191	28400
243	6	27692	28243	8	28293	None
244	7	27794	28350	5	28400	28640

under the first group of parameters reaches the maximum with 253, while the quantity of processing materials in the second process is still the minimum with only 211.

In order to more clearly see the operation of RGV in a shift, using matplotlib library of Python language to draw the changes of CNC involved in processing over time, which can also be regarded as the running route of RGV shown in Figure 2.

Figure 2. The RGV scheduling route under one procedure



As for two procedures schedule task, on the premise of satisfying the fastest and maximum for materials at the same time, the results are obtained by using the distance lanscape strategy algorithm designed in this paper for three groups of different parameters optimization. The tool types are shown in Table 9.

In this table, F denotes the tool used by CNC in procedure one, and S denotes the tool used by CNC in procedure two.

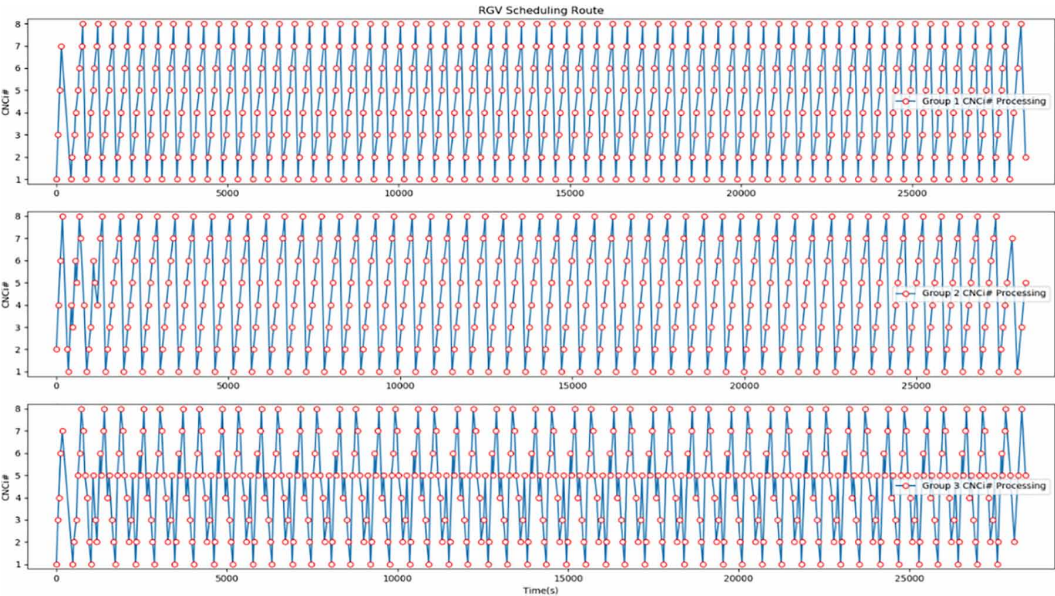
At the same time, according to the Algorithm 2, three different tool allocation scenarios are obtained for three groups of different parameters in Table 9. According to the tool allocation in the table and the scheduling routes in Figure 3, because the distribution of CNC in procedure one and two are different, the scheduling routes and execution cycles of RGV to perform loading and unloading tasks are also different.

It can be seen that the dynamic scheduling of RGV presents a very regular periodicity. Within a shift (8 hours), the density of parallel lines in the second group is small, and the density of parallel lines in the third group is large. This also confirmed that the number of CNC processed materials in the second group of parameters was less and the number of CNC processed materials in the third group was more. In addition, comparing the scheduling routes of one process under three different parameters, it can be found that the difference of parameters does not have much influence on the RGV scheduling route. The scheduling route of RGV is very similar to the internal disk scheduling algorithm of the computer.

Table 9. The tool types under three groups parameters

CNC i#	Tool Types		
	Group 1	Group 2	Group 3
CNC 1#	F	S	F
CNC 2#	S	F	S
CNC 3#	F	S	F
CNC 4#	S	F	F
CNC 5#	F	S	S
CNC 6#	S	F	F
CNC 7#	F	S	F
CNC 8#	S	F	S

Figure 3. The RGV scheduling route under two procedures



By observing the scheduling routes under three parameters, we can find that the first group of parameters is basically the same as the RGV scheduling routes under the second group of parameters, but because of the difference in the time of CNC executing the two procedures, the output of the finished material is also different. In addition, compared with the third group of RGV scheduling routes, it can be found that the tool allocation ratio of the first and second group is 3:5, while the tool allocation ratio of other two groups are 4:4, so the scheduling path has changed a lot.

5.3. Analysis of the Experimental Performance

A good scheduling model can shorten the production cycle, reduce the production costs, improve the production efficiency and other goals, and then improve the economic benefits of the system. The dynamic scheduling model of RGV directly affects the production capacity of the intelligent machining system. In order to analyze the performance of the intelligent scheduling strategy designed in this paper,

we introduce the expected number of processed materials, and calculate the total number of finished materials in the ideal state of CNC uninterrupted work (no waiting situation). Analysis the ratio of the actual number of processed materials to the expected number of processed materials and use it as the execution efficiency of the dynamic scheduling model to reflect the performance of the model.

From the Table 10 about the efficiency of the model with three parameters under one procedure, when finishing a material needs only one procedure, a large number of finished materials will be

Table 10. The execution efficiency of the RGV scheduling model under one procedure

Performance	Group 1	Group 2	Group 3
Actual quantity	382	359	392
Expected quantity	390	374	400
Execution efficiency	97.95%	95.99%	98.00%

processed by CNC with three sets of different parameters, and the execution efficiency will more than 95%.

From the Table 11 about the efficiency of the model with three parameters under two procedures, when finishing a material needs two procedures, the scheduling efficiency is higher only under the first group of parameters, and only about 75% in the other two groups.

Table 11. The execution efficiency of the RGV scheduling model under two procedures

Performance	Group 1	Group 2	Group 3
Actual quantity	253	211	244
Expected quantity	273	271	330
Execution efficiency	92.67%	77.86%	73.94%

The RGV dynamic scheduling model designed in this paper is simple and easy to understand, it can use the distance landscape strategy algorithm to obtain an ideal scheduling scheme, and quickly find the optimal allocation scheme of the quantity and position of each material, which shows the practicability and effective algorithm of this RGV dynamic scheduling model.

But at the same time, for the RGV scheduling model with one procedure, in the actual materials procession, there will be a time delay in the whole process of signal sending and receiving an action, which is not considered by the model. However, when the number of CNC or processing procedures increase, the solving time of the model will increase exponentially, so the effectiveness of the algorithm is not very strong.

In actual production, the delay time will be determined by a variety of random factors, so a fixed approximate delay time can be used to replace it and incorporate it into the calculation of the model will minimize the error. When the number of CNC increases, genetic algorithm, ant colony algorithm, and other approximate optimal solution algorithms can be used to calculate the tool distribution scheme and improve the computational efficiency.

6. CONCLUSION

In this paper, a distance landscape strategy based on the fitness landscape is used to solve the dynamic scheduling problem of RGV. With the development of our science and technology, the modern intelligent logistics system has been constantly improved. In order to make up for the problems of low efficiency and high maintenance cost exposed by general automation system and warehouse, the RGV dynamic scheduling model can be easily connected with other logistics systems to automatically transport, clean and process materials. In addition, it needs no human operation and runs fast. Thus, the workload of warehouse managers is significantly reduced, and labor productivity is improved. Meanwhile, the application of shuttle vehicles can make the logistics system very simple and convenient. A reasonable RGV scheduling strategy also provides more space for its development in industrial production, terminal sequencing, cloud scheduling system and other fields.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant No. 61903089) and the Science Foundation of Jiangxi University of Science and Technology (Grant No. JXXJBS18059).

REFERENCES

- Angel, E., & Zissimopoulos, V. (2001). On the landscape ruggedness of the quadratic assignment problem. *Theoretical Computer Science*, 263(1-2), 159–172. doi:10.1016/S0304-3975(00)00239-5
- Bouziri, H., Mellouli, K., & Talbi, E. (2011). The k-coloring fitness landscape. *Journal of Combinatorial Optimization*, 21(3), 306–329. doi:10.1007/s10878-009-9249-2
- Caamaño, P., Bellas, F., Becerra, J. A., & Duro, R. J. (2013). Evolutionary algorithm characterization in real parameter optimization problems. *Applied Soft Computing*, 13(4), 1902–1921. doi:10.1016/j.asoc.2013.01.002
- Daolio, F., Verel, S., Ochoa, G., & Tomassini, M. (2010, July). Local optima networks of the quadratic assignment problem. In *Proceedings of the IEEE Congress on Evolutionary Computation* (pp. 1-8). IEEE.
- Ferrara, A., Gebennini, E., & Grassi, A. (2014, November 1). Fleet sizing of laser guided vehicles and pallet shuttles in automated warehouses. *International Journal of Production Economics*, 157, 7–14. doi:10.1016/j.ijpe.2014.06.008
- Gagliardi, J. P., Renaud, J., & Ruiz, A. (2015, October 2). Sequencing approaches for multiple-aisle automated storage and retrieval systems. *International Journal of Production Research*, 53(19), 5873–5883. doi:10.1080/00207543.2015.1012600
- He, J., & Yao, X. (2005). A game-theoretic approach for designing mixed mutation strategies. In *Proceedings of the International Conference on Natural Computation*. Springer; . doi:10.1007/11539902_33
- Kallel, L., Naudts, B., & Reeves, C. R. (2001). Properties of fitness functions and search landscapes. In *Theoretical aspects of evolutionary computing* (pp. 175–206). Springer.
- Lee, S. G., De Souza, R., & Ong, E. K. (1996, October 15). Simulation modelling of a narrow aisle automated storage and retrieval system (AS/RS) serviced by rail-guided vehicles. *Computers in Industry*, 30(3), 241–253. doi:10.1016/0166-3615(96)00025-5
- Malan, K. M., & Engelbrecht, A. P. (2009). Quantifying ruggedness of continuous landscapes using entropy. In *Proceedings of the IEEE Congress on Evolutionary Computation CEC'09*. IEEE.
- Martina, C., Alessandro, P., & Fabio, S. (2018, January 1). Modelling of Rail Guided Vehicles serving an automated parts-to-picker system. *IFAC-PapersOnLine*, 51(11), 1476–1481. 10.1016/j.ifacol.2018.08.295
- Merz, P., & Freisleben, B. (2000). Fitness landscapes, memetic algorithms, and greedy operators for graph bipartitioning. *Evolutionary Computation*, 8(1), 61–91. doi:10.1162/106365600568103
- Müller, C. L., & Sbalzarini, I. F. (2011). Global characterization of the CEC 2005 fitness landscapes using fitness-distance analysis. In *Proceedings of the European Conference on the Applications of Evolutionary Computation*. Springer; . doi:10.1007/978-3-642-20525-5_30
- Prugel-Bennett, A., & Tayarani-Najaran, M. (2012). Maximum satisfiability: Anatomy of the fitness landscape for a hard combinatorial optimization problem. *IEEE Transactions on Evolutionary Computation*, 16(3), 319–338. doi:10.1109/TEVC.2011.2163638
- Reidys, C. M., & Stadler, P. F. (2001). Neutrality in fitness landscapes. *Applied Mathematics and Computation*, 117(2-3), 321–350. doi:10.1016/S0096-3003(99)00166-6
- Roy, D., Gupta, A., & De Koster, R. B. (2016, January 17). A non-linear traffic flow-based queuing model to estimate container terminal throughput with AGVs. *International Journal of Production Research*, 54(2), 472–493. doi:10.1080/00207543.2015.1056321
- Roy, D., Gupta, A., & De Koster, R. B. (2016, January 17). A non-linear traffic flow-based queuing model to estimate container terminal throughput with AGVs. *International Journal of Production Research*, 54(2), 472–493. doi:10.1080/00207543.2015.1056321
- Sáez, D., Cortés, C. E., & Núñez, A. (2008, November 1). Hybrid adaptive predictive control for the multi-vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy clustering. *Computers & Operations Research*, 35(11), 3412–3438. doi:10.1016/j.cor.2007.01.025

Shen, L., & He, J. (2010). A mixed strategy for evolutionary programming based on local fitness landscape. In *Proceedings of the 2010 IEEE Congress on Evolutionary Computation (CEC)*. IEEE.

Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding, and selection in evolution.

Zhu, Z., Xiao, J., He, S., Ji, Z., & Sun, Y. (2016, February 1). A multi-objective memetic algorithm based on locality-sensitive hashing for one-to-many-to-one dynamic pickup-and-delivery problem. *Information Sciences*, 329, 73–89. doi:10.1016/j.ins.2015.09.006

Wei Li is an associate professor of computational intelligence, evolutionary optimization, fitness landscape analysis, and large-scale optimization. He is a Senior Members of China Simulation Federation (CSF), China Computer Federation (CCF) and Intelligent Simulation Optimization and Scheduling Committee. He has held visiting professor positions at University of Calgary (2016). He is the steering committee chair of the annual Springer International Symposium on Intelligence Computation and Applications (ISICA) since 2015. He served as program and local arrangement chairs for 10+ international conferences.