RGV Dynamic Scheduling Strategy Based on Network Cellular Automaton and Marko Model

J Y Yin¹, N Chen¹, Z T Zhang² and S S Chen³*

¹ Department of Management, Wuhan University of Technology, Wuhan, Hubei, 430070, China
² These authors contributed to the work equally and should be regarded as co-first authors
³ Economic and Management School, Xidian University, Xi’an, Shanxi, 710126, China
³ Department of Science, Wuhan University of Technology, Wuhan, Hubei, 430070, China
*Corresponding author’s e-mail: chenshsh@whut.edu.cn

Abstract. In an intelligent processing system, if a Rail Guide Vehicle (RGV), also known as automated guide vehicle, is provided with the optimal dynamic scheduling strategy, the work efficiency and economic profit of the workshop can be greatly improved. This paper considers the variable number as well as the types of Computer Number Controller (CNC) and regards the real problem as the Job Shop Scheduling Problem (JSSP). To solve JSSP, this paper use Network Cellular Automaton (NCA) to abstract the flexible workshop system and apply the rule of one-step trial-and-error for the core link between the network cellular automaton and the Markov Transition Probability Matrix. Further this system is simulated using Markov chain in probabilities to expand the solution space with the optimal solution among according to the Monte Carlo experimental principle. Finally, the result can jump out of the local optimal solution and approximate the global optimal dynamic scheduling strategy by trial-and-error rule and Transition Probability Matrix. To verify the NCA-Markov model, this paper use an intelligent flexible workshop which produces two-process materials. By simulating this system with the model, the efficiency of the intelligent processing system can be greatly improved.

1. Introduction

The intelligent processing system is a human-machine integrated intelligent system that analyses, infers, judges, conceives and makes decisions by computer simulation of the intelligent activities of human experts. It plays an increasingly important role in the manufacturing industry. The intelligent processing system usually consists of multiple auxiliary devices: Computer Number Controller (CNC), Rail Guide Vehicle (RGV), RGV linear track, loading conveyor and unloading conveyor.

CNC is an automated machine tool with a program control system that can be machined by an automated control device set up in advance. RGV is a self-driving, smart car that can run freely on a fixed track[1]. CNCs are arranged on both sides, and a material conveyor belt is installed in front of each CNC. The loading conveyor is used to convey the raw material (unprocessed material) for the CNC, while the unloading conveyor is used to send the finished material (processed and cleaned material) out of the system. Others are auxiliary equipment that keeps the system up and running.
Figure 1 shows the specific process of RGV completing the loading and unloading of CNC. Figure 2 shows the position of the RGV, CNCs and conveyors in the intelligent processing workshop.

It’s important to study the dynamic scheduling strategy of RGV to optimize the efficiency of the intelligent processing system. So, Cellular Automaton is applied to this complex problem. Then this paper introduces a one-step trial strategy based on the State Transition Matrix of Markov Chain, which enhances the global search ability of the algorithm. At last, an example is used to verify the availability of the model.

2. Literature review

In recent years, there are three kinds of methods that can effectively solve JSSP: mathematical programming method, heuristic search algorithm and simulation model method.

The mathematical programming method is mainly based on the Branch & Bound Algorithm proposed by Balas[2]. Thereafter, Carlier et al.[3] proposed the pre-emptive scheduling (JPS) which is based on the Jackson’s Maximal Whole Remaining Time (MWR) Rule[4]. Davis et al.[5] proposed a decomposition strategy based on mathematical planning, they decomposed the scheduling problem into multiple sub-problems, thus reducing computational complexity.

There are many different approaches to solve JSSP by using Heuristic Search Algorithm. Brandimarte[6] chose existing dispatch rules to solve the path sub-problem adopting the idea of hierarchy, and then he used the Tabu Search Algorithm to solve sorting sub-problems. Xia and Wu[7] used a hybrid optimization approach based on Particle Swarm Optimization Algorithm and Simulated Annealing Algorithm to solve multi-objective JSSP problem. Gao J[8, 9], proposed a new Genetic Algorithm by combining Genetic Algorithm with local search and improving crossover and mutation operators. Pezzella[10] designed a Genetic Algorithm with better performance to solve JSSP by integrating different genetic strategies to generate the initial population.

The Simulation Model Method is to simulate the actual environment through modelling, which can include the factors that cannot be described by the mathematical formula to help determine the suitable scheduling method. For JSSP, Zhao F and Zhao Q[11] used Queuing Theory Model for simulation. Hao L F and Wang N[12] conducted simulation through Petri Net Model. Also, Cellular automaton can be used to the simulation of JSSP. With the help of the "bottom-up" modelling idea of Cellular Automaton (CA), the workshop operating system can be described by operating rules, which can effectively simulate complex systems.

Based on the researches above, this paper proposes a method combining Cellular Automation with Markov Chain to solve the specific problem of RGV scheduling in intelligent operating system. This paper selects NCA model to make closer to the simulation results and the actual scene, and introduces a one-step trial strategy. Through simulation iteration for many times, the approximate optimal solution of RGV dynamic scheduling problem can be obtained.
3. System loss
During a processing cycle, the CNC will wait for the RGV till it is available to process. The waiting time includes two possible situations which are waiting for loading to start working and waiting for unloading while holding the processed material. In these states, CNC’s machining capacity can’t be fully utilized, results in the loss of production capacity and economic waste. To measure the loss, the minimum of system loss is defined as the optimal target. The target function \( L \) will be denoted as

\[
\min L = \sum_{t=1}^{T} \sum_{i=1}^{N} L_i^t
\]  \hspace{1cm} (1)

In equation (1), \( L_i^t \) refers to the waiting time of the \( i^{th} \) CNC in time \( t \).

4. Network cellular automaton
Cellular automaton is a kind of grid dynamics model in which time, space and state are discrete with time causality and space interaction together affected its evolution. The basic components of cellular automaton are cellular space, neighbours, evolution rules, cells and their states. However, RGV is movable so that its neighbours will change as it moves. Therefore, this paper will use an improved network cellular automaton.

Network cellular automaton (NCA) is a general CA model[13], its brief defines are listed as followed:
- Cells are the nodes in the network rather than a grid or cubic in the dimensional space.
- Cellular types depend on the features of region where the cell belongs to.
- Cellular neighbours are selected followed the evolution rules. And it shouldn’t overstep the bounds of neighbours defined.

In conclusion, the JSSP is abstracted as into a network automaton model (NCA) composed of CNC, RGV and Flexible Job-shop scheduling rules. In this NCA model, the amount of cells is variable; cells are classified into the fixed type and movable type; the fixed cells are grouped by different processes in different regions; cells belonging to different regions have different neighbours; the type and amount of neighbour depends on the cellular states and evolution rules.

4.1. Division of the cellular space

4.1.1. Fixed cellular region. Every CNC in the job-shop is regarded as a fixed cell. And it is necessary to distinguish the types of CNC used for different process orders by neglecting their position in reality and putting the same type into a region. Each single region is considered as a network comprised of nodes representing CNCs.

4.1.2. Movable cell. The RGV is regarded as a movable cell. To simplify the question, the position of RGV will be neglected by describing its neighbours’ changing to show its movability. Due to the specificity of NCA, The RGV is regarded as a movable cell, whose type and amount of neighbour are uncertain and changeable.

4.2. Identification of cellular neighbors

4.2.1. Neighbors of the fixed cells. Firstly, as for a fixed cell CNC \( c_i \), other cells in the same region are its neighbours. Second, CNC will request RGV to unload when it finishes processing. However, when any CNCs used in the next process is available will RGV answer for the request, unloading the material and loading it on the next process. So its neighbours includes all cells in the region \( C_{i+1} \).

Simultaneously, according to the trial-and-error Rule, its neighbour includes all cells in the region \( C_{i+2} \).
Lastly, if the movable cell RGV is serving for it or at the request of it, the movable cell is also its neighbor.
Summarily, as for a fixed cell $c_{ij}$, $b$ refers to whether the movable cell RGV is serving for it or at the request of it. $b = 0$ means it isn’t. Then its neighbours are

$$\left[ c_{i1}, \ldots, c_{i(j-1)}, c_{i(j+1)}, \ldots, c_{iJ(i)} \right] \cup \left[ c_{(i+1)1}, \ldots, c_{(i+1)(J(i)+1)} \right] \cup \left[ c_{(i+2)1}, \ldots, c_{(i+2)(J(i)+2)} \right] \cup \left[ b \times C_m \right]$$

4.2.2. Neighbors of the movable cell. RGV can move along the rail, which implies the position of the movable cell is changeable. Therefore, the type and amount of its neighbour also changes as it moves. Defines that the neighbours of movable cell $C_m$ are what the CNC is serving for and what other CNC will be served in the next movement.

4.3. Cellular states

4.3.1. States of the fixed cells. In the production process of a flexible job-shop, there are three states of CNC, requesting for loading, working and requesting for unloading.

$$s_{ij} = \begin{cases} 
0 & \text{requesting for loading} \\
1 & \text{working} \\
2 & \text{requesting for unloading}
\end{cases}$$

4.3.2. State of the movable Cell. RGV moves and works according to the requests it received. What’s more, only when it finishes the request it is working for can it answer the next request. According to
this characteristic, it is defined that the value of its state $s_m$ to be the remaining time (second) that it will be available to move. To release the calculative load, the value of the state of the movable cell can be used as the time step.

4.4. Cellular evolution rules

4.4.1. Rule of mission response. When RGV receives loading request from region $C_i, 1 < i \leq n$, it should first check if there is a material finished on CNCs belongs to $C_{i-1}$. If there is none, the RGV will not answer requests until any material finishes process on CNC belonging to $C_{i-1}$.

4.4.2. Rule of proximity. Waiting in motionless, RGV may receive more than one request from different CNCs, which means conflict happens at this time. When RGV encounters conflict, it is necessary to provide strategy for it to make decision. Enlightened by the Greedy Algorithm, this paper prioritizes recent CNC requests. When requests are sent out from CNC from different distance, the nearest CNC will be served firstly.

4.4.3. Rule of trial-and-error. When RGV receives requests from the same distance, it will try one-step to all alternative decisions in simulation and calculate the corresponding system loss. Define a function between local system loss and probability $P(l)$ whose value decreases monotonically with the local loss $l$. Then the probability $p_i$ of the $i^{th}$ decision can be obtained by normalizing the values. For example, use the inverse proportional function to build the function $P(l)$ as below.

$$p_i = \frac{P(l_i)}{\sum P(l_j)}$$

$$P(l) = \frac{1}{l}$$

According to the idea of the Monte Carlo Simulation Experiment, with enough experimental times, the result will be close to the optimal solution. Also, different ways can be used to define the function, so that simulating in different probabilities more times to approach the optimal solution of the problem.

4.5. Global Constraints

According to the regulation of the flexible workshop, it is not allowed to process over 8 hours continuously in one day.

RGV moves along a certain length of track lined with CNCs on both sides. This paper ignores the position when RGV is moving and considers where it stops. In other word, this paper uses the position of the fixed CNC to show the position of RGV.

4.6. Evolution of the cellular automaton.

In conclusion, it is defined that the state of cellular space for arbitrary time is

$$S(t+1) = f(C, S(t), N, G)$$

In equation (4), $S$ represents the state of the cellular space and $S(t) = \{s_{11}(t), \cdots, s_{ij}(t), \cdots, s_{nj}(t), s_m(t)\}$. $f$ refers to the evolution rules. $C$ refers to the cellular space. $N$ refers to the neighbourhood and $G$ refers to the global constraints.
5. Transition probability matrix

A Markov chain is a Markov process in either discrete or continuous time with a countable state space [14]. It represents a random process in which the state space transitions from one state to another. Therefore, the CNC's work process can be regarded as a Markov process. The state transition probability matrix is proposed on the basis of the Markov chain, which expresses the nth result of some factors in a system where the transition is only affected by (n-1)th the result.

Using P(l) constructed above, the state transition probability for the CNC that issued the demand is obtained. If a total of K CNCs have issued a demand, then for the kth CNC, there are equations (3). A one-step transition probability matrix is built for the kth CNC by normalizing each line:

\[
P_k = \begin{bmatrix}
p_{11} & p_{12} & p_{13} \\
p_{21} & p_{22} & p_{23} \\
p_{31} & p_{32} & p_{33}
\end{bmatrix}
\]  

(5)

Among them, \( p_{ij}, 0 \leq p_{ij} \leq 1, \sum_{j=1}^{3} p_{ij} = 1, i, j = 1, 2, 3 \) indicates the probability that CNC transitions from i state at current time to j state at the next moment.

6. NCA-MARKOV model

As the construction of network cellular automaton this paper stated above, the matrix of state for each CNC is obtained. \( S(t) \) is used to refer to the state of the fixed cells CNCs in time t.

\[
S(t) = \begin{bmatrix}
s_1(t), s_2(t), \ldots, s_N(t)
\end{bmatrix}
\]  

(6)

This paper will try one step for each probable decision in simulation and get the Markov transition probability matrix in the next time step. According to Markov Chain’s definition, that is

\[
s(t+1) = s(t) \times P
\]  

(7)

So, the state matrix in the next time step \( s(t+1) \) and the matrix of cellular state in maximum time simulated by random is obtained.

\[
S = \begin{bmatrix}
S_1(0) & S_2(0) & \cdots & S_N(0) \\
\vdots & \vdots & \ddots & \vdots \\
S_1(t) & S_2(t) & \cdots & S_N(t) \\
\vdots & \vdots & \ddots & \vdots \\
S_1(t_{\text{max}}) & S_2(t_{\text{max}}) & \cdots & S_N(t_{\text{max}})
\end{bmatrix}
\]  

(8)

When there are enough experiments, multiple cellular state matrixes as the solution space can be obtained. Each solution space represents a scheduling plan with different amount of processed production. According to the Monte Carlo experiment, when enough randomized experiments are performed, the result will approach to a matrix of ultimate states. In other word, it is believed that under a large number of experiments, this model approaches the optimal solution step-by-step from local optimum to global optimum. That is, the optimal solution is in solution space of NCA-Markov model.

7. Model verification

7.1. System overview

Intelligent processing system is used in this part. The Figure 4 shows the positional relationship between eight CNCs and RGV, and the loading and unloading conveyor.
Figure 4. Workshop equipped intelligent processing system.

A set of system job parameter data provided in Table 1 is used.

Table 1. A set of data sheet for intelligent processing system operating parameters.

| System Operation Parameters | Data (seconds) |
|-----------------------------|----------------|
| The time required for RGV to move 1 unit | 20 |
| The time required for RGV to move 2 units | 33 |
| The time required for RGV to move 3 units | 46 |
| The time required for CNC to complete the first process | 400 |
| The time required for CNC to complete the second process | 378 |
| The time for RGV to load and unload on CNC1#, 3#, 5#, 7# | 28 |
| The time for RGV to load and unload on CNC2#, 4#, 6#, 8# | 31 |
| The time required for RGV to complete a cleaning operation | 25 |

Note: 8 hours of continuous operation per shift.

7.1.1. System operation flow.
1. After the intelligent processing system is powered on, the RGV is in the initial position between the CNC1# and CNC2#, and all the CNCs are in idle state.
2. Usually, if a CNC is in idle state, it will send a loading request to the RGV; otherwise, an unloading request is sent to the RGV immediately after the machining operation is completed.
3. After the RGV completes operation for a CNC, robot arm will rotate and the clinker of one robot will move to the cleaning tank for cleaning operation (only the processed material is cleaned).
4. RGV immediately determines the execution of the next job instruction after completing a task. At this time, if no other work command is received, the RGV waits in place until the next work request. After a CNC completes the processing task of a material, it immediately sends a demand signal to the RGV. If the RGV fails to arrive at it immediately, the CNC will wait. Facilities layout simulation.

According to the data of the Table 1 and the system operation flow, the following situations are solved by using the model to judge the validity of the model.

The machining process that each CNC is responsible for is shown in the Figure 5, where black circle represents the CNC for the first process and white circle represents the CNC for the second process.

Figure 5. Position of CNCs used in different processes.
7.2. Analysis of results of two-process RGV scheduling model

According to the material processing operation and the data provided in the Table 1, this paper uses the NCA-Markov model to solve the problem and the results are showed in Table 2.

Table 2. Results of two-process RGV scheduling model.

| Result                  |       |
|-------------------------|-------|
| Amount of products(units)| 253   |
| Operation efficiency    | 85.56%|

The amount of products obtained from the set of data is 253 units. It can be seen that in the case of two-process operation, the system operation efficiency of the model reaches 85.56%, which proves that the model in this article can effectively optimize the dynamic scheduling of RGV.

Table 3. Results of two-process compared with general greedy algorithm.

| Result                |       |
|-----------------------|-------|
| NCA-Markov model(units)| 253   |
| Greedy algorithm(units)| 242   |

Also the greedy algorithm is used to compare the results with those calculated by NCA-Markov model, as shown in the Table 3. The results obtained by the simulation model are also better than the results of the greedy algorithm. For the first set of results, this model still enables the intelligent processing system to have a higher operational efficiency, while the results of the second and third groups are not satisfactory because the number of CNCs responsible for the two processes fails to match the time required for the two-step process. Therefore, this simulation model can still obtain better results even if it is under abnormal conditions.

Figure 6. RGV’s moving trajectories in two-process scheduling.

Observing the RGV movement trajectory diagram of the two-step process shown in the Figure 6, it’s obvious that for two-process material processing, the movement of RGV is frequent and complex, but it has cyclic feature on the whole. By analysing the specific data of moving tracks simulation, the data at some positions do not change with the period and have variation, which proves that the one-step trial-and-error rule plays a role. That is, the movement of RGV conforms to the expectation of the model.

Figure 7. Gantt chart in two-process scheduling.

Figure 7 shows the working Gantt chart of 8 CNCs obtained according to the simulation results. It’s seen that the scheduling of RGV in the model make full use of the CNC, and the processing sequence of products is very compact. The idle time of each CNC is reduced to a low level. At the same time, the processing time of each product on CNC is close to the actual processing time of the
product. It indicates that this model improves the processing efficiency of the system and is close to the ideal machining state.

8. Conclusion
According to the flexibility and dynamics of the RGV scheduling problem, the network cellular automaton model is applied to establish the cellular state matrix. Then the motion process is clearly described by applying the Markov possibility transition matrix. Also, a one-step trial-and-error strategy with probability is adopted. Based on Monte Carlo experimental principle, this model has gradually approached the global optimum under enough experiments and finally reached the overall coverage. In other words, it can significantly improve the efficiency of the intelligent processing system of the workshop using this model for simulation compared with the traditional greedy algorithm model.

References
[1] Lacomme, P., Larabi, M., and Tchernev, N. (2013) Job-shop based framework for simultaneous scheduling of machines and automated guided vehicles. International Journal of Production Economics, 143: 24-34.
[2] Balas, E. (1969) MACHINE SEQUENCING VIA DISJUNCTIVE GRAPHS: AN IMPLICIT ENUMERATION ALGORITHM. Operations Research, 17: 941-957.
[3] Carlier, J., and Pinson, E. (1990) A PRACTICAL USE OF JACKSON’S PREEMPTIVE SCHEDULE FOR SOLVING THE JOB SHOP PROBLEM. Annals of Operations Research, 26: 269-287.
[4] Jackson, J. R. (1955) Scheduling a Production Line to Minimize Maximum Tardiness.
[5] Davis, W., and Jones, A. (1988) A real-time production scheduler for a stochastic manufacturing environment. International Journal of Computer Integrated Manufacturing, 1: 101-112.
[6] Brandimarte, P. (1993) Routing and scheduling in a flexible job shop by tabu search. Annals of Operations Research, 41: 157-183.
[7] Xia, W., and Wu, Z. (2005) An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems.
[8] Jie, G., Gen, M., Sun, L., and Zhao, X. (2007) A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems. Systems Engineering, 53: 149-162.
[9] Jie, G., Sun, L., and Gen, M. (2008) A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. Computers & Operations Research, 35: 2892-2907.
[10] Pezzella, F., Morganti, G., and Ciaschetti, G. (2008) A genetic algorithm for the Flexible Job-shop Scheduling Problem. Computers & Operations Research, 35: 3202-3212.
[11] Zhao, F., and Zhao, Q. (2013) A hybrid flow shop scheduling model based on queuing theory and its performance analysis. Journal of Computational Information Systems, 9: 249-256.
[12] Hao, L. F., and Wang, N. (2016) A neural network modeling method using fsRNA-GA for 2-dimensional overhead crane. Chongqing: International Conference on Mechanics and Mechanical Engineering: 515-525.
[13] Schroeder, W. J., Lorensen, W. E., Montanaro, G. D., and Volpe, C. R. (1992) VISAGE: an object-oriented scientific visualization system, In Conference on Visualization.
[14] Meyn, S., Tweedie, R. L., and Glynn, P. W. (1999) Markov Chains and Stochastic Stability: Mud maps.