RGV Scheduling Model for Single-Process Intelligent Machining Systems

Authors: Weigang Zhou, Feng Qianqian, Zhou Weigang

Date: 2019-07-16T00:00:00+00:00

Abstract

This paper studies the scheduling problem of rail-guided vehicles (RGV) in a single-process intelligent machining system. This problem formed part of Question B in the 2018 National Undergraduate Mathematical Modeling Contest. The system consists of a rail-guided vehicle (RGV) and several computer numerical control (CNC) machines, among other components. The RGV controls multiple CNCs to process multiple workpieces, and the RGV scheduling scheme determines the efficiency of the system. A mathematical model for the problem is presented with the RGV's movement path as decision variables, the operation completion times of the RGV on CNCs as time nodes, and the remaining processing time of workpieces as state variables. However, some parameters in the model use decision variables as subscripts. By defining new variables and constraints, the model is transformed into a nonlinear mixed-integer programming model without variable subscripts or piecewise functions. Finally, numerical examples are provided to demonstrate the correctness and operability of the model.

Full Text

RGV Scheduling Model for Intelligent Processing System with One Process

FENG Qianqian, ZHOU Weigang*, WU Yuanhong, HE Guanghui, CHEN Shijun
School of Mathematics and Statistics, Hubei University of Arts and Science, Xiangyang 441053

Abstract

This paper studies the scheduling problem of an intelligent processing system involving a Rail Guided Vehicle (RGV). This problem originates from Problem

B of the 2018 China Undergraduate Mathematical Contest in Modeling. The system comprises a Rail Guided Vehicle (RGV), multiple Computer Numerical Control (CNC) machines, and other components. The RGV manages multiple CNCs to process multiple workpieces, and the RGV scheduling scheme determines the overall system efficiency. A mathematical model is developed with the RGV' s moving path as decision variables, the operation completion times on CNCs as time nodes, and the remaining processing time of workpieces as state variables. However, some parameters in this model have decision variables as subscripts. By introducing new variables and constraints, the model is reformulated to eliminate decision-variable subscripts and piecewise functions, transforming it into a nonlinear mixed-integer programming model. Finally, a numerical example is provided to demonstrate the correctness and operability of the model.

Keywords: intelligent machining system; scheduling problem; nonlinear mixed integer programming; mathematical modeling

---

Introduction

The intelligent processing system under consideration consists of a Rail Guided Vehicle (RGV) and several Computer Numerical Control (CNC) machines. The RGV controls multiple CNCs to complete the processing of multiple workpieces, and the RGV scheduling scheme determines the efficiency of the system. This problem was part of the 2018 National Undergraduate Mathematical Modeling Competition [1], and reference [2] notes that it is a highly practical and operational problem.

The system can be described as follows: The RGV is an unmanned intelligent vehicle that runs freely on a fixed track. It can automatically control its moving direction and distance according to instructions and is equipped with a robotic arm. The front end of the RGV' s robotic arm has two grippers that can sequentially grab one workpiece each through rotation to complete loading, unloading, and cleaning operations. At any given time, the RGV can only perform one of the following tasks: moving, waiting, loading/unloading, or cleaning. Each CNC is equipped with identical cutting tools, and workpieces can be processed on any CNC. The processing requires only one operation. A conveyor belt supplies raw workpieces (unprocessed) to the CNCs and transports finished workpieces (processed and cleaned) out of the system. Only the time for RGV movement, loading, unloading, and cleaning operations is considered in determining system efficiency. For a more detailed description of the system, see reference [1].

Reference [2] first analyzes the time consumption of the RGV' s operating path in one cycle, assigns numbers to workpieces, and uses 0-1 decision variables to indicate whether a workpiece is assigned to a particular CNC. Based on the minimum time-consuming path, a nonlinear mixed-integer programming model is established and solved using a designed algorithm. This paper presents a different modeling approach from reference [2]. Reference [3] studied this problem using simulation methods, while reference [4] investigated the shortest RGV operating path in one cycle using dynamic programming.

1.1 Symbols and Analysis

The processing of a workpiece involves: loading, CNC processing, unloading, and finally cleaning. Both loading, unloading, and cleaning require RGV operations. After unloading, the RGV can either clean the workpiece first or, using its other gripper, complete the loading operation for another workpiece before cleaning. It is evident that completing another workpiece' s loading first saves more time. One RGV simultaneously controls multiple CNCs. The time required for the RGV to move from CNC i to CNC j is denoted as d_{ij} (distance). Each workpiece' s processing time is p (process), and cleaning time is w (wash). When $p \leq w$, only one CNC needs to work, so we assume the loading time is a (arrive) and unloading time is l (leave). Other operation times are negligible.

On a CNC that is not occupied by a workpiece, the RGV performs one loading operation. On an occupied CNC, the RGV performs unloading first, then loading, and finally cleaning. The consecutive operations of unloading, loading, and cleaning on the same CNC are recorded as one operation. Let $z_i = 0$ denote loading on an idle CNC, and $z_i = 1$ denote unloading, loading, and then cleaning on a CNC that has finished processing. For n workpieces, there correspond n loading operations. After the moment when no more raw workpieces are taken, there are still workpieces being processed on CNCs. When the RGV arrives at each CNC, the required operations are: unloading and then cleaning finished workpieces. The RGV scheduling sequence can follow the first-processed-first-unloaded order. After completing an operation on one CNC, the RGV immediately begins moving to the next CNC. For model simplicity, we do not consider the processing after the moment when no more raw workpieces are taken, and we aim to minimize the time span from when the RGV starts working until the completion of the n-th operation.

The RGV' s initial position is $y_0 = 0$. The n operations in chronological order correspond to n positions of the RGV, denoted as the RGV' s moving path $y = (y_1, y_2, \dots, y_n)$. The period from the operation end time on CNC y_{i-1} to the operation end time on CNC y_i is called stage i. In stage i, the RGV first moves from CNC y_{i-1} to CNC y_i, then performs operation z_i on the workpiece at CNC y_i. If the workpiece processing on CNC y_i is not yet complete when the RGV arrives, the RGV must wait.

The state vector of CNCs at the beginning of stage i (i.e., the end time of stage $i-1$) is denoted as $h^{i-1} = (h_1^{i-1}, h_2^{i-1}, \dots, h_m^{i-1})$, where $h_j^{i-1} \geq 0$ indicates that at the end of stage $i-1$, CNC j has a workpiece with remaining processing time h_j^{i-1}; $h_j^{i-1} = 0$ indicates that at the end of stage $i-1$, CNC j has no workpiece, corresponding to RGV operation $z_i = 0$; $h_j^{i-1} = -p$ indicates that at the end of stage $i-1$, there is a finished workpiece waiting on CNC j, corresponding to RGV operation $z_i = 1$. The initial time is 0, and the initial state is $h^0 = (-p, -p, \dots, -p)$. For stage i, given state h^{i-1} and RGV position y_{i-1}, the decision is to determine the RGV' s next position y_i and operation z_i. Since the decision y_i is determined by state h^{i-1}, $y = (y_1, y_2, \dots, y_n)$ is the only independent decision.

1.2 Stage Duration

When the RGV arrives at a CNC, it performs corresponding operations based on the CNC' s status. For example, when the RGV arrives at CNC y_i, if the CNC is idle, the RGV only loads the CNC, as shown in the second row of Table 1 . Cases where the RGV arrives at CNC y_i and there is a workpiece on the CNC are shown in the third and fourth rows of Table 1.

Table 1: CNC Status and RGV Operations

CNC Status	RGV Operation
Remaining processing time > 0	Wait → Unload → Load → Clean
Remaining processing time $= 0$	Unload → Load → Clean

From Table 1, the duration of stage i, denoted as π_i, can be expressed as:

$$\pi_i = \begin{cases} d_{y_{i-1}y_i} + a, & z_i = 0 \\ d_{y_{i-1}y_i} + l + a + w, & h_{y_i}^{i-1} \le p, z_i = 1 \\ d_{y_{i-1}y_i} + l + a + w, & h_{y_i}^{i-1} > p. \end{cases}$$

1.3 State Transition Formulas

For CNC j: - If $j \neq y_i$, meaning CNC j has no operation, then $h_j^i = \max\{0, h_j^{i-1} - \pi_i\}$. If $h_j^{i-1} = 0$, CNC j still has no workpiece; if $h_j^{i-1} > 0$, the workpiece on CNC j has been processed for an additional π_i time units.

- If $j = y_i$ and $z_i = 0$, meaning CNC j has no workpiece and the operation is loading, after loading is completed, the remaining processing time of the workpiece on CNC j is p, i.e., $h_j^i = p$.
- If $j = y_i$ and $z_i = 1$, the operation is unloading, loading, and then cleaning. After the operation is completed, the remaining processing time of the workpiece on CNC j is p.

The state at the end of stage i is:

$$h_j^i = \begin{cases} \max\{0, h_j^{i-1} - \pi_i\}, & j \neq y_i \\ p, & j = y_i, z_i = 0 \\ p, & j = y_i, z_i = 1. \end{cases}$$

1.4 Model

The model with minimum time length as the objective (Model 1) is:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} \pi_i \\ \text{s.t.} \quad & \pi_i = \begin{cases} d_{y_{i-1}y_i} + a, & z_i = 0 \\ d_{y_{i-1}y_i} + l + a + w, & h_{y_i}^{i-1} \leq p, z_i = 1 \\ d_{y_{i-1}y_i} + l + a + w, & h_{y_i}^{i-1} > p \end{cases} \\ & h_j^i = \begin{cases} \max\{0, h_j^{i-1} - \pi_i\}, & j \neq y_i \\ p, & j = y_i, z_i = 0 \\ p, & j = y_i, z_i = 1 \end{cases} \\ & h_j^0 = -p, \quad j = 1, 2, \ldots, m \\ & y_0 = 0, \quad y_i \in \{1, 2, \ldots, m\}, \quad i = 1, 2, \ldots, n \\ & z_i \in \{0, 1\}, \quad i = 1, 2, \ldots, n. \end{aligned}$$

2 Model Transformation

Since models containing variable subscripts and piecewise functions cannot be directly solved by many software packages, we modify Model 1 below.

For equation (1), let $x_{ij} \in \{0,1\}$ satisfy $\sum_{j=1}^{m} x_{ij} = 1$, where $x_{ij} = 1$ indicates that the RGV' s operation in stage i is on CNC j. The decision changes from y to $X = (x_{ij})$. Define $e_i^1, e_i^2, e_i^3 \in \{0,1\}$ satisfying $e_i^1 + e_i^2 + e_i^3 = 1$, corresponding to the three parts of equation (1). The constraint $h_{y_i}^{i-1} \leq p$ is equivalent to $h_j^{i-1} \leq p + M(1 - x_{ij})$, where M is a sufficiently large integer. Let ε denote an arbitrarily small positive number. The constraint $h_{y_i}^{i-1} > p$ is equivalent to $h_j^{i-1} \geq p + \varepsilon - M(1 - x_{ij})$. Equation (1) is modified to:

$$\pi_i = \sum_{j=1}^{m} \left[e_i^1 (d_{ij} + a) + e_i^2 (d_{ij} + l + a + w) + e_i^3 (d_{ij} + l + a + w) \right] x_{ij}.$$

For equation (2), define $o_{ij}^1, o_{ij}^2 \in \{0,1\}$ with $o_{ij}^1 + o_{ij}^2 = 1$. Similar analysis can modify equation (2) to:

$$h_j^i = \sum_{i=1}^{m} \left[o_{ij}^1 \max\{0, h_j^{i-1} - \pi_i\} + o_{ij}^2 p \right] x_{ij}.$$

From (3) and (4), Model 1 is transformed into Model 2:

$$\begin{aligned}
\min \quad & \sum_{i=1}^{n} \pi_i \\
\text{s.t.} \quad & \pi_i = \sum_{j=1}^{m} \left[e_i^1 (d_{ij} + a) + e_i^2 (d_{ij} + l + a + w) + e_i^3 (d_{ij} + l + a + w) \right] x_{ij} \\
& h_j^i = \sum_{i=1}^{m} \left[o_{ij}^1 \max\{0, h_j^{i-1} - \pi_i\} + o_{ij}^2 p \right] x_{ij} \\
& \sum_{j=1}^{m} x_{ij} = 1, \quad i = 1, 2, \dots, n \\
& \sum_{k=1}^{3} e_i^k = 1, \quad i = 1, 2, \dots, n \\
& \sum_{k=1}^{2} o_{ij}^k = 1, \quad i, j = 1, 2, \dots, m \\
& h_j^{i-1} \le p + M(1 - x_{ij}) + M(1 - e_i^2), \quad i, j = 1, 2, \dots, m \\
& h_j^{i-1} \ge p + \varepsilon - M(1 - x_{ij}) - M e_i^3, \quad i, j = 1, 2, \dots, m \\
& h_j^0 = -p, \quad j = 1, 2, \dots, m \\
& x_{ij}, e_i^k, o_{ij}^k \in \{0, 1\}.
\end{aligned}$$

---

3 Solution

For Model 1, if the decision $y = (y_1, y_2, \dots, y_n)$ is determined, we can start from the first stage and iteratively calculate the values of π_i and h_j^i using equations (1) and (2). The objective function value can be quickly computed, and the optimal solution under the assumption of "fixed RGV moving path in each cycle"can also be quickly obtained from Model 1. Assuming the RGV' s moving path is fixed in each cycle, we calculated the third set of data from the example in reference [1] using MATLAB. The end time of 380 stages was 7.9236 hours. Reference [2] reported completing 393 workpieces within 8 hours. Since our model does not consider the processing of the last few workpieces, there were still 8 workpieces being processed on CNCs at the end time of 380 stages. Therefore, our results are basically consistent with those reported in reference [2]. From the values of π_i and h_j^i, the start times of loading/unloading and other operations can be easily derived.

For Model 2, we used LINGO to compute examples with only 2 to 5 workpieces, and the results matched those from Model 1. Because Model 2 contains many nonlinear constraints, the computational speed is relatively slow. Designing faster algorithms for Model 2 is a topic requiring further research.

---

4 Conclusion

The selection of time nodes and state variables is crucial for establishing Model 1. Model 1 does not require prior consideration of the time consumption of one RGV cycle, and the approach of ignoring the last few workpieces makes the model more concise. When a CNC fails, it can be removed from the CNC set, the other CNCs maintain their current states, and the model is re-solved. After the failed CNC is repaired, it can be added back to the CNC set with its state defined as idle, while other CNCs maintain their current states, and the model is re-solved. When processing times are very short, the optimal solution may involve some CNCs remaining idle. The multi-operation problem can also be considered based on Model 1.

Funding: This research was supported by the Scientific and Technological Research Fund of Hubei Provincial Education Department (D20162602).

References

[1] 2018 China Undergraduate Mathematical Contest in Modeling Problem [EB/OL]. [2019-2-15].

[2] Han Zhonggeng. Review of 2018 China Undergraduate Mathematical Contest in Modeling: Dynamic Scheduling Strategy for Intelligent RGV [EB/OL]. [2019-2-15].

[3] Zhong Zhuohui, Gao Yutong, Zhou Jiaxin, Li Chun. Dynamic Scheduling Strategy for Intelligent RGV [J]. Computer Science and Application, 2019, 9(1): 89-95.

[4] Wo Yunting, Yu Yitong. Intelligent Scheduling Strategy for RGV Based on Dynamic Programming Algorithm [J]. Industrial Technology Innovation, 2018, 5(6): 44-47+52.

Note: Figure translations are in progress. See original paper for figures.