HAL open science

Dynamic Weight Configuration of Dispatching Rule Using Machine Learning

Jong-Ho Shin, Chaekyo Lee, Sangrae Kim, Jun-Gyu Kang

To cite this version:

Jong-Ho Shin, Chaekyo Lee, Sangrae Kim, Jun-Gyu Kang. Dynamic Weight Configuration of Dispatching Rule Using Machine Learning. IFIP International Conference on Advances in Production Management Systems (APMS), Aug 2018, Seoul, South Korea. pp.110-115, 10.1007/978-3-319-99704-9_14 . hal-02164853

HAL Id: hal-02164853

https://inria.hal.science/hal-02164853v1

Submitted on 25 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Dynamic weight configuration of dispatching rule using machine learning

Jong-Ho Shin[1[0000-1111-2222-3333]] , Chaekyo Lee[1], Sangrae Kim[2], and Jun-Gyu Kang[2[1111-2222-3333-4444]]

[1] Chosun University, Gwangju 61452, Republic of Korea
[2] Sungkyul University, Anyang 14097, Republic of Korea
jun-gyu.kang@sungkyul.ac.kr

Abstract. The manufacturing execution systems (MES) is one of the key elements consisting smart factory. It is responsible for shop floor control by performing managing resources, dispatching production orders, executing production orders, collecting production data, analyzing production performances, and so on. Through these functionalities, the MES aims high productivity. The dispatching in the MES helps these aims. The selection of job in manufacturing execution systems (MES) is performed by dispatching rule. The dispatching rule is composed of several factors affecting scheduling objective and constraint. In most cases, the dispatching rule is expressed as the weighted sum of factors and the weight moderates the relative importance among factors. To find optimal weight configuration requires heavy calculation burden so that it cannot adapt dynamic order changes. To solve this problem, one of machine learning algorithms is used in this study. The multi-layer perceptron learns the best weight configuration according to orders and predict the best weight configuration for new orders. The proposed method is tested by field data and proved its usefulness.

Keywords: Dispatching Rule, Weight Configuration, Multi-Layer Perceptron.

1 Introduction

The manufacturing execution systems (MES) is one of the key elements consisting smart factory. As a part of the smart factory on shop floor level, the MES performs various roles such as managing resources, dispatching production orders, executing production orders, collecting production data, analyzing production performances, and so on. Through these functionalities, the MES aims to achieve high productivity and reduced cycle-time. Especially, the dispatching and executing are closely related with these aims. Therefore, lots of methodologies to improve the dispatching and execution are developed and implemented into the MES.

The most common method to improve the dispatching is to use dispatching rules in the selection of job for execution. The dispatching rule is a kind of construction heuristic algorithm to achieve optimal scheduling. The dispatching rule priorities all the possible jobs awaiting for processing in front of machine. Whenever a machine becomes

available, the dispatching rule calculates priorities of all the waiting jobs and selects the highest priory job in order to process it. According to objectives and constraints of scheduling, the dispatching rule is composed of various factors and, mostly, it is expressed as a weighted sum of these factors. The relative importance among factors are coordinated by the weights which are multiplied to the factors. In case that several factors are involved in the dispatching rule, it is difficult to find appropriate weights since the relative importance among factors is vague. To find an optimal weight configuration needs heavy calculation time. Moreover, the dynamically changing orders make job selection for optimal scheduling complicate. The weight configuration also should be modified according to the dynamic order changes. Hence, it is necessary to develop a methodology to define proper weight configuration with respect to orders.

To cover these problems, this study proposes a dynamic weight configuration methodology according to orders using machine learning technology. In the proposed methodology, the most appropriate weight configuration which can give the lowest tardiness and the highest throughput can be decided whenever order information is given. To do this, the simulation test to find the best weight configuration is performed and the found best weight configuration depending on orders through simulation is learned by the machine learning algorithm. Then, the machine learning algorithm predicts and provides the best weight configuration for the given new orders without simulation test. The proposed methodology is tested by the real manufacturing process and proves its effectiveness.

2 State-of-the-art

Regarding scheduling problems, many research works have proposed multi-attribute dispatching rules. Korytkowski *et al.* (2013) proposed an ant colony optimization to determine dynamic multi-attribute dispatching rules to maximize job shop system performance. Yang et al. (2007) uses genetic algorithm to solve a multi-attribute combinatorial dispatching (MACD) decision problem in a flow shop with multiple processors (FSMP) environment. Rokni and Fayek (2010) applied fuzzy set theory in a multi-criteria optimization framework for industrial shop scheduling.

The scheduling performance of multi-attribute dispatching rule depends on its scaling factor values (Pfund et al, 2008). Jeong and Randhawa (2001) proposes multi-attribute dispatching rules for automatic guided vehicle. In this work, three attributes are included in the dispatching rule and additive waiting model is used to compute weights. A neural network is also used to adjust weights reflecting changes in system.

The dispatching rule using multi attribute is widely used in the field. However, the balance of relative importance remains difficult problem. Especially, the dynamic weight configuration requires heavy calculation time so that it is required to relieve this problem.

3 Multi-Layer Perceptron(MLP) for weight configuration

3.1 Dispatching rule of targeted stage

The studied shop floor in this research produces the spark plug of automobile. The manufacturing process of the spark plug is composed of several stages. During this process, this study focuses on dispatching improvement at the assembly stage. The assembly stage has parallel machines and the waited jobs in front of machines is selected by the dispatching rule. The information required in the calculation of the dispatching rule is collected by MES and transferred from ERP system. Then, the MES calculates dispatching values of each waiting job, selects the best one, and guides operators to work with the most appropriate job.

The dispatching rule used in this stage has six factors, which is expressed as follows.

$$D = w_1f_1 + w_2f_2 + w_3f_3 + w_4f_4 + w_5f_5 + w_6f_6 \quad (1)$$

Each factor from f_1 to f_6 is included in order to keep due date, make group with same kind of jobs, give special priority to OEM job, avoid machine idle, follow specific machine allocation, and better throughput. For each waiting job, the dispatching rule value '*D*' is calculated using equation (1) and the lowest value of job is selected to be processed. The value of factors are extracted from order information and the monitored data from MES. The weights from w_1 to w_2 plays important role to moderate the relative importance among factors. Depending on the characteristics of orders, these weights should be modified so as to assure the lower tardiness and high throughput.

3.2 Experimental environment

For the development and test in this study, the simulation environment is developed and used. The following figure shows developing environment.

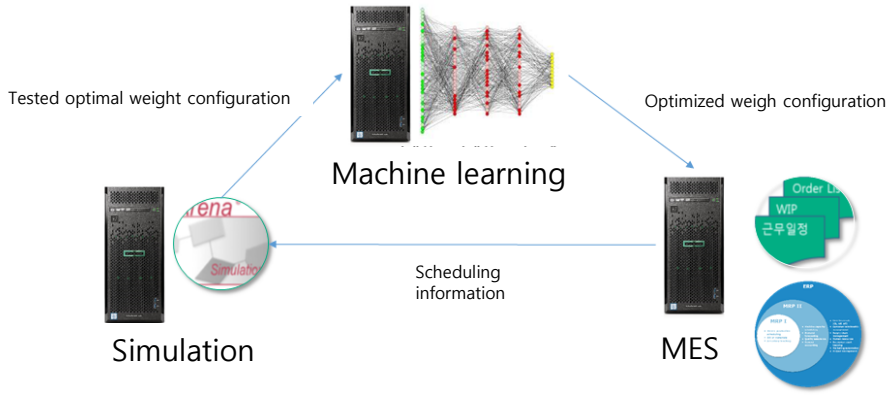

Fig. 1. Experimental architecture and data.

The information of orders required for scheduling such as orders, work-in-process (WIP), working schedule, and so on is collected and provided to simulation model. The simulation model is made by Arena simulation software and simulates the whole manufacturing process. The dispatching rule is implemented in the simulation model and used whenever each machine becomes available in order to select job. The simulation module tests all the possible combinations of weight configuration with given orders and, then, finds the best weight configuration. The best weight configuration with respect to orders is learned by machine learning algorithm. The learned machine learning algorithm is implemented to moderate weight configuration according to orders in the field.

3.3 Experiment and result

To train machine learning algorithm, the input and output should be defined. The input of scheduling is defined by orders. The orders have various information such as product ID, processing route, order date, due date, and so on. This information is converted as a vector which has more than 270 elements. Depending the orders for scheduling, a new vector is defined and each orders can be specified. The output is extracted from simulation module. Among, the tested possible weight configurations, the best weight configuration giving lower tardiness and higher throughput is used as the output of machine learning. For training, several pairs of input and output are prepared.

The used machine learning algorithm in this study is multi-layer perceptron (MLP). The multi-layer perceptron is a class of feedforward artificial neural network and used to distinguish non-linear data and is used for classification and regression. The MLP is composed of several layers consisting of perceptrons. In this study, the MLP has 20 layers and the same number of perceptrons as input vector per each layer.

To conduct the experiment, we generate 2000 input and output sets. Among 2000 test data sets, 70 % of data is used for training, 15% of them is used for validation, and the rest is used for testing. The performance of the MLP model is expressed as root mean square error (RMSE) between actual the best weight configuration by simulation and the predicted weight configuration by MLP. The MLP in this study is coded and tested using MATLAB 2017b.

The following table shows the experimental results.

Table 1. Performance comparison experiment depending on training function.

Model	Training Function	RMSE	Error Max	Error Min
1	Levenberg-Marquardt backpropagation	0.192	1.163	0.046
2	Bayesian regularization backpropagation	0.133	1.961	0.071
3	Scaled conjugate gradient backpropagation	0.265	0.877	0.079
4	Resilient backpropagation	0.264	1.0054	0.065

As shown in Table 1, four kinds or training function are tested with the fixed MLP structure. Bayesian regularization backpropagation as Model 2 shows the lowest RMSE. However, this function generates the predicted weight configuration with large

errors. The RMSE of Levenberg-Marquardt backpropagation as Model 1 has large error than Model 2 but the number of prediction values with a large error value is less.

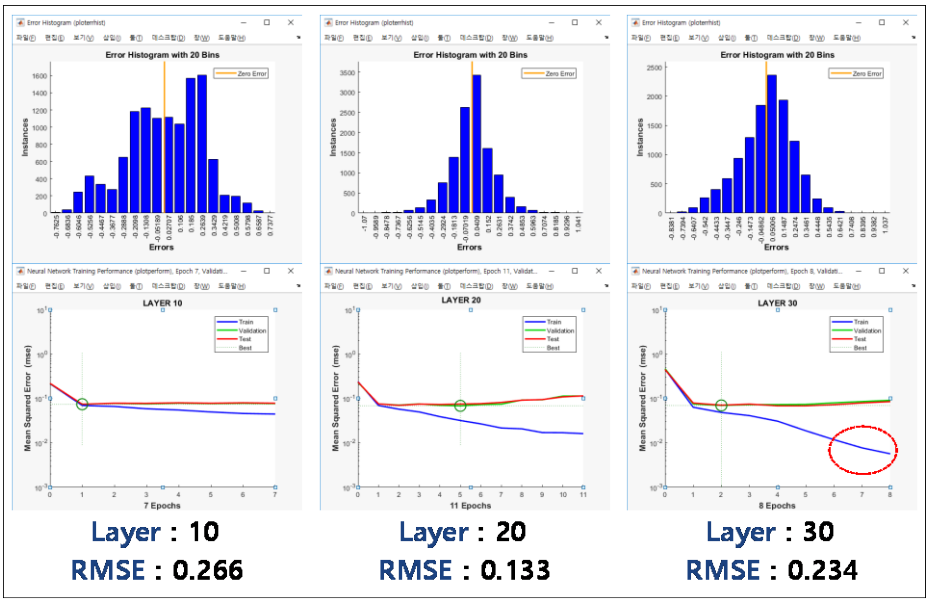

Fig. 2. Model performance comparison by layer

Fig. 2 shows the error histogram change with respect to MLP structure. As errors between real value and predicted value, the histogram which has more instances near zero means better MLP structure. According to Fig. 2, MLP structures having layer 10 and 30 give worse prediction result than MLP with 20 layers. In addition, when the layer is 30, the red circle shows that the RMSE of the training data becomes low. This means that, as the layers of the Model get deeper, the over-fitting phenomenon occurs so that the MLP is optimized for training data and does not predict test data well.

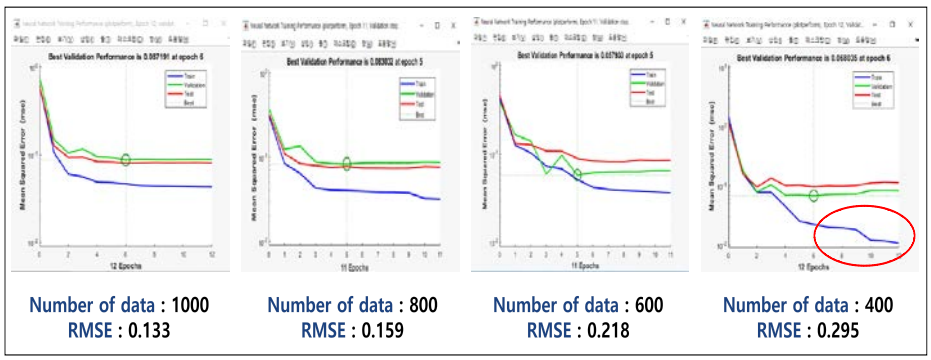

Fig. 3. Comparison of model performance by number of data

Fig. 3 shows the effect of data used in machine learning. When the data is less, the overfitting occurs and the RMSE becomes worse. Hence, to get usable result, an appropriate amount of data is needed.

4 Conclusion and discussion

In many shop floor, the MES plays an important role to manage jobs. The dispatching rule is one of key method to priority jobs and execute job in the MES. The dispatching rule is composed of multiple factors and they are balanced by weights. In dynamic order environment, the relative importance of factor should be adapted according orders. In this study, we have proposed an intelligent methodology to configure weight of dispatching rule. The proposed methodology uses multi-layer perceptron to learn the best weight configuration. The training data for learning is generated by simulation module. The experiments based on real field problem is performed and the effectiveness of MLP is validated through experiments.

References

1. Jeong, B.H., Randhawa, S.U.: A multi-attribute dispatching rule for automated guide vehicle systems. International Journal of Production Research 39(13), 2817–2832 (2001).
2. Korytkowski, P., Rymaszewski, S. & Wiśniewski, T.: Ant colony optimization for job shop scheduling using multi-attribute dispatching rules. International Journal of Advanced Manufacturing Technology 67, 231–241 (2013).
3. Yang, T., Kuo, Y., Cho, C.: A genetic algorithms simulation approach for the multi-attribute combinatorial dispatching decision problem. European Journal of Operational Research 176(3), 1859–1873 (2007).
4. Pfund, M., Fowler, J.W., Gadkari A., Chen, Y.: Scheduling jobs on parallel machines with setup times and ready times. Computers and Industrial Engineering 54(4), 764–782 (2008).
5. Rokni, S., Fayek, A. R.: A multi-criteria optimization framework for industrial shop scheduling using fuzzy set theory. Integrated Computer-Aided Engineering 17(3), 175–196 (2010).