Dynamic Job Shop Scheduling Based on Order Remaining Completion Time Prediction

Hao Wang[1], Tao Peng[1](✉), Alexandra Brintrup[2], Thorsten Wuest[3], and Renzhong Tang[1]

[1] Institute of Industrial Engineering, School of Mechanical Engineering, Zhejiang University, Hangzhou 310027, China
tao_peng@zju.edu.cn

[2] Institute of Manufacturing, Department of Engineering, University of Cambridge, Cambridge CB3 0FS, UK

[3] Industrial and Management Systems Engineering, West Virginia University, Morgantown, WV 26506-6070, USA

Abstract. Emerging ubiquity of smart sensing in production environments provide opportunities to make use of fine-grained, real-time data to support decision-making. One, currently untapped opportunity is the prediction of order remaining completion time (ORCT) which can be used to improve production scheduling. Recent research has focused on the development of ORCT prediction models however, their integration into scheduling algorithms is an understudied area, especially in job shop environments where processing times can be highly variable. In this paper, an artificial neural network was developed to predict ORCT based on real-time job shop status data which is then integrated with classical heuristic rules for facilitating dynamic scheduling. A simulation study with four scenarios was developed to test the performance of our approach. The results demonstrated improved completion time, however tardiness was not reduced under all scenarios. In moving this research forward, we discuss the need for further research into combining static and dynamic characteristics and priority rule design for satisfying multiple objectives.

Keywords: Order remaining completion time prediction · Dynamic scheduling · Artificial neural network · Heuristic rules

1 Introduction

Many small- and medium-sized enterprises (SMEs) focusing on Make-To-Order (MTO) production face the challenge of providing on-time order delivery whilst operating under highly uncertain environments [1]. In a typical job shop environment, simple First-Come-First-Serve production leads to long completion times and order tardiness. Order remaining completion time (ORCT), also termed as order completion time [2] or job remaining time [3], is defined as the time that elapses between a customer's order is created and order completion in the production environment [4]. The actual order due date

© IFIP International Federation for Information Processing 2022
Published by Springer Nature Switzerland AG 2022
D. Y. Kim et al. (Eds.): APMS 2022, IFIP AICT 664, pp. 425–433, 2022.
https://doi.org/10.1007/978-3-031-16411-8_49

depends on the ORCT which impacts the efficacy of subsequent production scheduling. However, predicting the ORCT of an incoming or in-production order is quite challenging, particularly in a job shop with high product variance and production uncertainties such as fluctuating processing time and machine availability.

Due to the highly variant nature of job shops, heuristic rule-based dynamic scheduling has been implemented to create feasible schemes in near real-time. In the past decades, shortest processing time rule (SPT), earliest due date rule (EDD), least work content in the next queue rule (WINQ) and cost over time rule (COVERT) [5] have been proposed to prioritize the waiting jobs considering characteristics including processing time, lead time of jobs, and work content of machines. Once the ORCT of order at any time can be acquired, it can be considered as an important indicator for evaluating the order's priority for dynamic scheduling. Dynamic scheduling involves using these priority rules which then changes the processing order of the waiting jobs, which in turn affects their completion times. Given the interdependence of ORCT and production scheduling, there is a gap in literature exploring how these solution approaches may be combined and what would the resulting efficacy of doing so would be.

This paper explores this research gap by proposing to dispatch incoming and in-production orders based on online predicted ORCT. Firstly, production related data that will affect ORCT are analyzed and used to train an artificial neural network based prediction model (ORCTpNet). Secondly, orders are prioritized based on predicted ORCT for to minimize order completion time and tardiness.

2 Brief Review

The essence of our proposed approach relies in ORCT prediction based on real-time job shop production data. We first review existing works on ORCT prediction, mainly focusing on the use of machine learning based approaches. Next, research on heuristic rules based dynamic scheduling are introduced.

2.1 ORCT Prediction

In ORCT prediction studies have mainly concentrated on analytical methods [6, 7] by constructing mathematical models under restrictive assumptions or simulation methods [8, 9] that require long computing time to obtain sufficient samples. The asumptative nature of analytical models, and high computational times required by simulation models prohibit these from being used in real-time production scheduling.

With the increased popularity of smart sensing solutions deployed in job shop environments, production data can be continuously streamed which may be used to develop ANN models, creating a data-driven approach. In an ANN model data from machinery, WIP, materials and orders can be feature-engineered as inputs to estimate ORCT. Compared with analytical and simulation based methods, a data-driven method treats a complex manufacturing system as a black box and has the advantage of achieving higher accuracy without having to resort to restrictive assumptions. However, ANN also has the limitation of a lack of explainability and high dependency on data quality.

Researchers have highlighted many factors to be considered to predict ORCT successfully. According to Liu [10], the key features affecting ORCT include production process, buffer queue length, order content, workload and status of machines, continuous working period of machines as some of these. Huang et al. [11] developed a hybrid approach combining the Long Short-Term Memory (LSTM) network and analytical system model to predict the product completion time in a flow shop considering the product type, processing time and queue length of machines, but no dynamic events were considered. Wang and Jiang [2] proposed a deep neural network-based order completion time prediction method by using real-time job shop RFID data, which include the types and waiting list information of all WIPs, real-time processing progress of all WIPs under machining, but the breakdown of machines was not considered. In the real production environment, dynamic events such as machine failure and repair, random arrival of orders and worker shift and their impacts on the ORCT should be considered, in order to make the prediction models in accordance with the actual situations.

2.2 Dynamic Scheduling with Heuristic Rules

Dynamic scheduling does not create or update schedules. Instead, it uses heuristic rules to prioritize jobs waiting for processing. When a source becomes available, the scheduling algorithm sorts the job queue by pre-designated criteria with the first-ranked job being dispatched for processing. In real-world scenarios, construction heuristics are often the method of choice for creating feasible schedules of reasonable quality within a relatively short computation time, as fast decision-making processes are of vital importance in uncertain manufacturing environments [12]. In Blackstone's work [13], frequently used heuristic rules for scheduling were classified into four categories: processing time-oriented, due date-oriented, workload-oriented and combined rules. EDD and COVERT rules both suggest to take the due date into consideration when evaluating the priority of a waiting job, but it's challenging to provide an accurate due date of an incoming job, and an experience-dependent estimated due date, for example, the Little's law [14], make these heuristic rules fail to achieve the desired outcome.

Gyulai et al. [15] investigated a flow shop scheduling problem to adjust the job priorities based on machine learning-based prediction of manufacturing lead times. To achieve situation-awareness, dynamic scheduling should rely on the combination of static data (e.g., product attributes) and event-driven data. Therefore, online ORCT prediction should be considered simultaneously in dynamic scheduling to allow for more realistic scheduling solutions.

3 Research Methodology

In this study, a discrete MTO manufacturing job shop is investigated where each machine can process a certain task of each job. A job will be processed on each machine once and some machines may not be occupied by the job's task sequence. Each task of a job can only be processed on one dedicated machine. A simulation model has been developed to reflect the real production and act as a test bed for the proposed research framework (as shown in Fig. 1). The framework outlines four main tasks.

Task 1: production data from sensors and devices are statistically analyzed to mathematically describe the uncertainties.
Task 2: a simulation model reflecting the real situations and dynamic events is built and verified upon statistic data.
Task 3: the ORCTpNet is developed and trained based on hybrid simulation and production data.
Task 4: heuristic rules-based dynamic scheduling with online predicted ORCT is implemented in the simulation to reduce tardiness and order completion time.

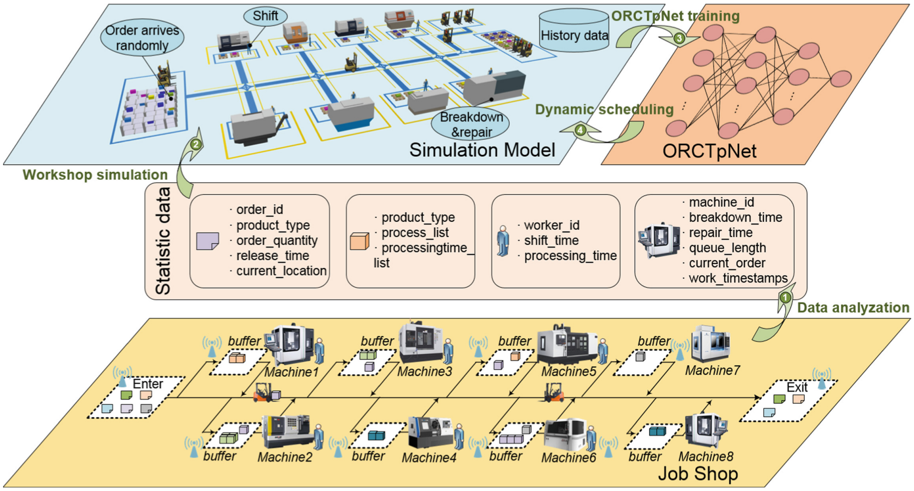

Fig. 1. Framework of dynamic scheduling with online predicted ORCT.

3.1 Problem Statement

There are five products to be processed on a same eight-machine production line and each of them has individual routes and processing times. Orders are released randomly to the job shop and the due date d_j of order j can be calculated by Eq. (1) where OR_{jt} is the predicted ORCT of order j at its release time r_j. For a given amount of orders O, C is the completion time of the last finished order as Eq. (2) and F is the sum of tardiness of all orders as Eq. (3). The objective functions of interest include the completion time C and tardiness F, depicted as Eq. (4).

$$d_j = r_j + OR_{jt}, t = r_j, j = 1, 2, \ldots, O \tag{1}$$

$$C = \max C_j, j = 1, 2, \ldots, O \tag{2}$$

$$F = \sum\nolimits_{j=1}^{O} \max(C_j - d_j, 0) \tag{3}$$

$$f = \min(C, F) \tag{4}$$

3.2 ORCTpNet Training

When an order is released to the job shop or enters the waiting buffer of one machine, there are numerous production status data to be considered to predict the ORCT. Theoretically, if an order can be processed without delays at each machine, the ORCT would simply be the sum of processing times, determined by product type and quantity. In real production, the order might need to wait in the queue of an occupied or faulty machine and the processing times might fluctuate. Here, four types of production status data are selected to construct the feature set of ORCTpNet: order-related, product-related, worker-related and machine-related data as shown in Fig. 1. Product and order-related data for a given location can be acquired from ERP, whilst the remaining data are real-time and streamed from MES. By tracing back the completion time of historical orders and the work timestamps of each machine, the order remaining completion time at each machine can be calculated to train ORCTpNet as Algorithm 1.

Algorithm 1. ORCTpNet training

```
Input: Dataset $D$, train_set_size $s$, epochs $E$
    $D$ = MinMaxScaler($D$)
    train_set, test_set = split($D$, $s$)
    construct ORCTpNet $A(\odot)$
    repeat for $A(\odot)$
            for epochs in 1,…, $E$ do
          train $A(\odot)$ with train_set
    examine $A(\odot)$ with test_set
    until for the performance of $A(\odot)$ is satisfactory
    Save the trained $A^*(\odot)$
Output: $A^*(\odot)$
```

3.3 Close-Loop Production Scheduling

Numerous open-source tools have been introduced to develop and train machine learning models such as TensorFlow, Scikit-learn and PyTorch. These tools are usually implemented in Python and need to integrate with job shop environment constructed by simulation platform. Incompatibility between different programming languages induces time delays. In this paper, the connection between Java-based job shop simulation model and Python-based ORCT prediction model is realized through flask-http-client [16]. When an order enters in the waiting buffer of a machine, a request is triggered by the machine to acquire the ORCT of this order. As a response, the predicted ORCT is sent to the machine. Then, the priority of the incoming order can be calculated and the waiting orders in the queue are reordered. In this way, online ORCT prediction and close-loop dynamic scheduling can be realized with minimal time delays and instability.

4 Simulation Experiments

4.1 Parameter Setting

We investigate an office furniture manufacturing job shop. The standard processing time of each process is normally distributed with mean 12.0 min and variance 0.5. Workers change shifts 8 h one day. Order arrival follows a triangularAV distribution with the mean value of 15 min and variance of 0.5, and the quantity of product in each order follows a uniform (50, 100) distribution. The time between machine failures follows a uniform (5.0, 8.0) hour distribution and the repair time follows a triangular (0.5, 2.0, 1.0) hour distribution.

4.2 Model Training and Performance Evaluation

During the simulation, each order follows an individual route and its processing time on each machine is related to product type, quantity and the operator. When the machine breaks down, the order being processed is suspended until the machine is repaired. After model logic and trial validation, the simulation model has been run for 30 days and 36801 records were obtained to train the ORCTpNet. Each record contains the production status data when the order was entered in the waiting buffer of the machine as input features and the actual ORCT was the label. In order to meet the near-real-time requirements of online prediction, the structure of the neural network should be as simple as possible such that computational time is minimized whilst prediction accuracy maximised. A normal back-propagation network including four dense layers (100-100-50-1 units) is deployed on TensorFlow and the output of each layer is processed by batch normalization to accelerate the training and avoid over-fitting [17]. The optimizer used in this case study is AdamOptimizer with learning rate α = 0.001 and the training batch size is 128. The training process is depicted in Fig. 2. Both training loss and testing loss decreases substantially with the training epochs.

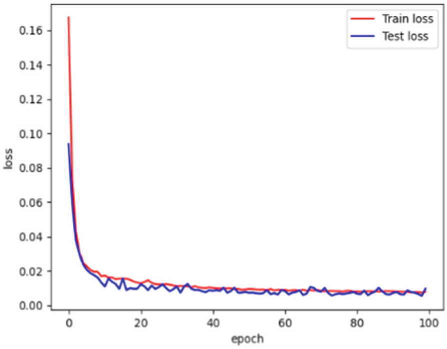

Fig. 2. Training process.

When ORCTpNet has been used at different machines, the results were satisfactory of the first four machines, whereas the prediction results were always zero for the last four machines as shown in Fig. 3. The reason was that orders were about to be completed at the last four machines so the ORCT at these locations was smaller than that of the machines forward, which led to sample values close to zero after Min-max normalization. To overcome this problem, the model was retrained by using samples from the last four machines and the resulting two networks were deployed to predict ORCT at different locations.

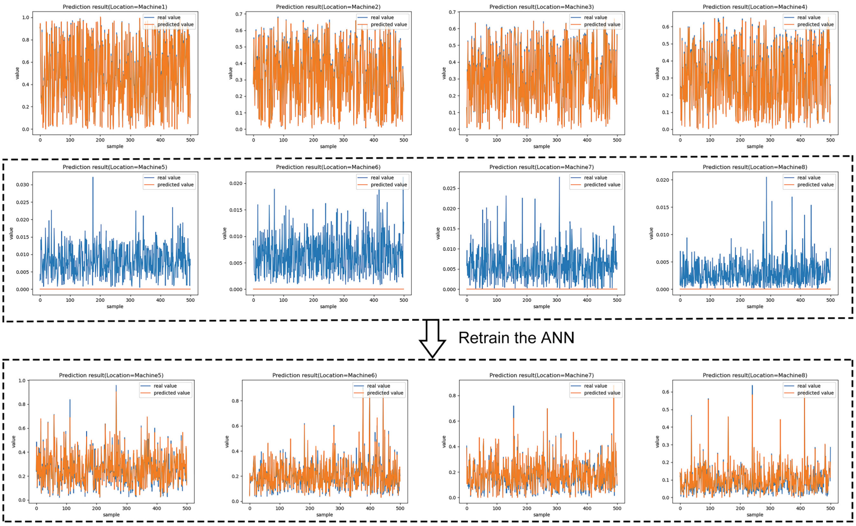

Fig. 3. ORCT prediction results at different machines.

4.3 Performance Compared with Classic Heuristic Rules

The aforementioned classic heuristic rules such as EDD and COVERT consider a fixed order due date d_j and remaining processing time R_{jm} to calculate the priority of job j in the queue of machine m at time t (ρ_{jmt}), whereas in a priority-based production environment, the expected due date is continuously fluctuating with preempt-production at each machine. We introduce the real-time ORCT $ORjt$ to calculate order due date at any time, then the EDD and COVERT rules can be adapted as Eq. (5) and (6).

$$\mathrm{EDD}^* : \rho_{jmt} = 1/(t + OR_{jt}) \tag{5}$$

$$\mathrm{COVERT}^* : \rho_{jmt} = \frac{1}{p_{jm}} \max[1 - \frac{\max\left(OR_{jt} - R_{jm}, 0\right)}{k_1 R_{jm}}, 0] \tag{6}$$

Four scenarios were developed to validate our approach, which differed in their order arrival interval λ in minutes, total order quantity O and product quantity Q (Scenario 1: $\lambda =$ triangularAV(2.5, 0.5), $O = 50$, $Q =$ uniform(20, 50); Scenario 2: $\lambda =$ triangularAV(7.5, 0.5), $O = 50$, $Q =$ uniform(20, 50); Scenario 3: $\lambda =$ triangularAV(2.5, 0.5),

O = 100, Q = uniform(20, 50); Scenario 4: λ = triangularAV(2.5, 0.5), O = 50, Q = uniform(50, 100)). The results are shown in Table 1. Overall, SPT and WINQ rules hold a relatively low completion time but heavy tardiness. EDD and COVERT rules have good performance on tardiness reduction. The introduction of real-time ORCT to replace the fixed due date had a positive effect in completion time reduction, but the tardiness was not always reduced. The experimental results demonstrate that the consideration of real-time ORCT is somehow beneficial for the objective of completion time reduction, however, a well-designed combined heuristic rule that considers both the static and dynamic characteristics should be developed to satisfy the two objectives.

Table 1. Experimental results of heuristic rule-based scheduling.

Rule		Scenario 1	Scenario 2	Scenario 3	Scenario 4
SPT	*C*	31.30	34.67	61.80	95.72
	F	50.79	144.24	342.65	910.27
WINQ	*C*	34.80	33.57	61.84	94.32
	F	121.05	110.82	209.03	484.13
EDD	*C*	34.78	34.72	64.66	96.75
	F	60.35	99.85	61.52	78.82
EDD*	*C*	33.13	32.31	62.04	94.42
	F	57.42	64.14	237.47	421.74
COVERT	*C*	36.59	37.48	66.70	93.04
	F	89.39	81.62	104.43	272.08
COVERT*	*C*	33.98	34.69	63.93	92.90
	F	71.06	92.21	107.94	462.75

5 Conclusions and Outlook

This paper proposed an ANN-based approach to predict real-time ORCT in a dynamic job shop such that uncertainties stemming from processing times and availability of machines may realistically inform subsequent scheduling decisions. In our approach, first hybrid simulation data and production data including information of orders, products, workers and machines were feature-engineered as inputs of the ORCT prediction model. To achieve the objectives of minimal completion time and tardiness, the predicted ORCT was considered as an indicator for evaluating order priority, such that a situation-aware closed-loop production control approach could be created. Experimental results showed a positive effect in completion time reduction but tardiness was not always reduced.

Although our results demonstrate the usefulness of ORCT prediction in scheduling, feature extraction and fusion can be further conducted to analyse which variables are more influential in deterring ORCT. Simulation data can be replaced by the real data from factories to train the ANN model, and new technologies such as data cleaning and pre-processing can be employed to support practical application. For manufacturing industries, seasonal demands lead to diverse distribution of the training samples, which necessitates a robust model for prediction accuracy. Finally, a composite priority rule considering static and dynamic order characteristics can also be developed to have a good performance in all aspects.

References

1. Bender, J., Ovtcharova, J.: Prototyping machine-learning-supported lead time prediction using AutoML. Procedia Comput. Sci. **180**(5), 649–655 (2021)
2. Wang, C., Jiang, P.: Deep neural networks based order completion time prediction by using real-time job shop RFID data. J. Intell. Manuf. **30**(3), 1303–1318 (2017). https://doi.org/10.1007/s10845-017-1325-3
3. Fang, W., Guo, Y., Liao, W., et al.: Big data driven jobs remaining time prediction in discrete manufacturing system: a deep learning-based approach. Int. J. Prod. Res. **58**(9), 2751–2766 (2020)
4. Gunasekaran, A., Patel, C., Tirtiroglu, E.: Performance measures and metrics in a supply chain environment. Int. J. Oper. Prod. Manag. **21**(1), 71–87 (2001)
5. Wang, H., Peng, T., Tang, R., et al.: Smart agent-based priority dispatching rules for job shop scheduling in a furniture manufacturing workshop. In: ASME 2020 15th International Manufacturing Science and Engineering Conference, pp. 1–8. Virtual Online (2020)
6. Altendorfer, K., Jodlbauer, H.: An analytical model for service level and tardiness in a single machine MTO production system. Int. J. Prod. Res. **49**(6), 1827–1850 (2011)
7. Hu, S., Zhang, B., Zhang, X.: Order completion date estimation and due date decision under make-to-order mode. Ind. Eng. J. **15**(3), 122–129 (2012)
8. Li, M., Yang, F., Wan, H., et al.: Simulation-based experimental design and statistical modeling for lead time quotation. J. Manuf. Syst. **37**, 362–374 (2015)
9. Hsieh, L., Chang, K., Chien, C.: Efficient development of cycle time response surfaces using progressive simulation metamodeling. Int. J. Prod. Res. **52**(9–10), 3097–3109 (2014)
10. Liu, D., Guo, Y., Huang, S., et al.: A SOM-FWFCM based feature selection algorithm for order remaining completion time prediction. China Mech. Eng. **32**(9), 1073–1079 (2021)
11. Huang, J., Chang, Q., Arinez, J.: Product Completion time prediction using a hybrid approach combining deep learning and system model. J. Manuf. Syst. **57**, 311–322 (2020)
12. Braune, R., Benda, F., Doerner, K., et al.: A genetic programming learning approach to generate dispatching rules for flexible shop scheduling problems. Int. J. Prod. Econ. **243**, 108342 (2022)
13. Blackstone, J., Phillips, D., Hogg, G.: A state-of-the-art survey of dispatching rules for manufacturing job shop operations. Int. J. Prod. Res. **20**(1), 27–45 (2007)
14. Little, J.D.: OR FORUM---Little's law as viewed on its 50th anniversary. Oper. Res. **59**(3), 536–549 (2011)
15. Gyulai, D., Pfeiffer, A., Bergmann, J., et al.: Online lead time prediction supporting situation-aware production control. Procedia CIRP **78**, 190–195 (2018)
16. Flask-http-client. https://pypi.org/project/flask-http-client/
17. Loffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. JMLR.org (2015)