Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 81 (2019) 701–706

www.elsevier.com/locate/procedia

52nd CIRP Conference on Manufacturing Systems

Machine learning technologies for order flowtime estimation in manufacturing systems

Rory Murphy[a], Anthony Newell[a], Vincent Hargaden[a], Nikolaos Papakostas[a]*

[a]*Laboratory for Advanced Manufacturing Simulation, School of Mechanical and Materials Engineering, University College Dublin, Belfield, Dublin 4, Ireland*

* Corresponding author. Tel.: +353-1-716-1741; *E-mail address:* nikolaos.papakostas@ucd.ie

Abstract

The problem of order due date assignment is an important issue for many companies and especially SMEs, which typically rely on production managers' best estimates to assign customer order due-dates. This paper investigates the use of machine learning (ML) technologies for order flowtime estimation in dynamic job shops, utilising a discrete event simulation framework for modelling manufacturing operations. The data generated via simulation is used by a series of ML technologies for predicting when orders could be completed. A series of experiments are conducted, and the performance of the proposed approach is compared with conventional due date assignment methods.

Keywords: scheduling; order promising; dynamic due date assignment

1. Introduction

The analysis of the complex operation of modern manufacturing systems in the context of networked production has been receiving increasing attention in the recent years. These operations include activities such as planning, order release, scheduling, control and dispatching. Machine Learning's (ML) core technologies align well with the complex problems which manufacturers face and can be applied to reduce labour costs, product defects, unplanned downtimes, improve transition times and production speed [1].

To remain competitive in today's global markets, modern manufacturing systems, aside from providing high quality products, competitive pricing, and reduced lead times, must also meet promised due-dates, the specified date when production of an order is expected to be complete. Flowtime estimation is the problem of forecasting the timespan between an order arriving in a manufacturing system and its departure. Reliable due-date assignment is critical in a job shop as it affects both customer relations and shop floor management practices. A good due-date assignment (DDA) method (DDAM) can perform timely delivery and reduce inventory costs, thereby enhancing customer satisfaction and company's competitive advantage. However, accurate flowtime estimation is a challenging problem in a dynamic environment, due to the stochastic nature of arriving jobs and the uncertainty of processing times and sequences caused by some dispatching rules [2].

This paper focuses on ML techniques as a viable option for manufacturing flowtime estimation in a dynamic job shop. Predictive analytics are increasingly important to Supply Chain Management however, many SMEs (small and medium enterprises) do not possess the infrastructure, technical capability or financial means for their implementation [3]. Flowtime estimation is important for managing shop floor operations in manufacturing networks. When this information is shared downstream with partners, it allows customers to improve their inventory control operations, especially if used in tandem with demand forecast information [4]. Increasingly, companies are employing available to promise (ATP) and capable to promise (CTP) systems within their IT landscapes. These systems allow companies to connect customer orders

10.1016/j.procir.2019.03.179

with resources in the organization, thereby providing customers with due-date estimates, which, if accurate, may lead to improved supply chain performance [4].

The estimation of job flowtimes has been an important issue since the late 1960s. Earlier studies focused on the application of simple, regression-based rules for flowtime estimation. Some conventional rules, e.g. TWK, SLK, NOP, JIQ, WIQ, JIS, WIS and PPW, were proposed by different researchers and a thorough comparison was conducted [5]. Their study showed that a combination of job characteristics and shop status information should be incorporated to develop DDA rules, that the choice of dispatching rules influences the shop and due-date performance, and that information about the shop congestion along a job's routing is more useful than information about general shop conditions.

Dynamic Total Work Content (DTWK) based on the TWK rule and Dynamic Processing Plus Waiting (DPPW) based on the PPW rule are two dynamic DDA rules presented in [6] and are capable of adjusting dynamically the flowtime estimation by using feedback information on current shop load conditions. These rules were developed by applying Little's Law [7] and queuing theory. The results of this study demonstrate that these dynamic DDA rules provide better results than their static counterparts. DTWK and DPPW are defined by equations (1) and (2) respectively, where d_i, r_i, denote the due date and arrival time of job i, where N_{s_t}, λ, μ_p, μ_g denote the number of jobs in the system, mean interarrival time of jobs, the mean operation processing time and the average number of operations per job respectively. N_{q_t} and m_j represent the total number of jobs in the queues of each machine and the number of operations of job i. p_{ij} then represents the processing time of the job i for operation j.

$$d_i = r_i + \max\left[1, \frac{N_{s_t}}{\lambda\mu_p\mu_g}\right]\sum_{j=1}^{m_j} p_{ij} \tag{1}$$

$$d_i = r_i + \frac{N_{q_t}\, m_j}{\lambda\mu_g} + \sum_{j=1}^{m_j} p_{ij} \tag{2}$$

Conventional DDA rules rely on linear or low-order nonlinear regression methods and lack the ability to reveal complex nonlinear relationships between actual job flowtimes and relevant information concerning job characteristics and system state [2]. To overcome these shortcomings researchers, since the early 1990s, have been applying ML and artificial intelligence (AI) methods to deal with DDA problems. These include the use of artificial neural networks (ANNs) [8-11], regression trees and data mining methods [12-14], heuristics and genetic programming methods [15, 16] The results of these studies showed that the AI models provided superior performance compared to conventional and regression-based methods.

Simulation-based approaches have also been proposed, utilizing decision-making principles in a number of industrial applications, requiring, however, the development of accurate and often complex simulation models [17, 18, 19, 20].

In this paper five ML models using ANNs, gradient boosting, data mining methods and regression trees with ensemble and boosting concepts, are evaluated across three dispatching policies for a flow shop and hybrid flow shop model. A discrete-event simulation model is first built to imitate the production process of a highly dynamic flow shop. More complex dynamic shop behaviour such as rework, machine downtime, machine set-up time, and machine repair time is considered. A series of simulation runs are then conducted with varying dispatching rules (FCFS, SPT, EDD) to collect data for training, validating and testing the ML models. Experiments are carried out to evaluate the trained models' performance in terms of Mean Absolute Lateness (MAL), Root Mean Squared Lateness (RMSL) and Fill Rate (FR) criteria. An experiment is first carried out for a batch flow process where order batches are not split and flow as a single unit. The second experiment was carried out for the case where jobs flow as individual units. A preliminary comparison is made between the ML methods tested and the dynamic DDAMs: DTWK and DPPW. These dynamic DDAMs were developed for job shop flowtime estimation under different operating conditions and were not designed to predict order flowtime estimation in complex manufacturing processes.

The paper is organised as follows. Section 2 outlines the five ML models used for due date assignment. Section 3 introduces the experimental design and data generation. Section 4 provides the experimental results and gives a brief analysis. The conclusions and suggestions for future work are presented in Section 5.

2. Research methodology

A selection of estimation methods was implemented to allow for comparison between ML algorithms in terms of accuracy, computational speed and relative ease of implementation. These methods include the Cubist datamining method, two ANN methods, h2o and neuralnet, random forest regression using randomForest R, which utilises ensemble learning of regression trees, and a gradient boosting method XGBoost R. The objective of these methods is to train a model on labelled data. A set of input variables is fed to the model to produce an estimate of the dependent variable. The models are therefore regression-based, supervised learning models, which are trained to predict the flowtime of an order. The input variables are obtained from the simulation model along with the corresponding real flowtime values and are comprised of job characteristics and shop condition at the time of arrival for an order. All models are developed using the R language and environment for statistical computing and graphics [21] and are trained with the same seed number (used to initialise the pseudorandom number generator) to allow for comparisons between models. The input variables used to train the models are provided in Table 1.

3. Experimental design

A discrete-event simulation model was implemented in Java using the DESMO-J (Discrete Event Simulation and Modelling in Java) framework [22].

Table 1. Selected attributes describing job and shop characteristics.

Main Class	Sub Class	Number of Features
Order/Job Characteristics	Number of Operations	1
	Batch Size	1
	Total Work Content	1
	Processing Time at Each Workstation	5*, 9**
	Processing Sequence	1
Shop Conditions	Moving Average of Inter-Arrival Time of Orders	1
	Total Number of Jobs in System	1
	Remaining Work Content of All Jobs in System	1
	Moving Average of Inter-Departure Times (Jobs)	1
	Moving Average of Inter-Departure Times (Orders)	1
	Average Order Flowtime (Last 3 Orders)	1
	Number of Jobs in Queues Along Order's Routing	1
	Remaining Work Content of All Jobs in Queues Along Order's Routing	1
	Remaining Work Content at Each Workstation Along Order's Routing	1
	Number of Jobs in Each Workstation's Queue	5*, 9**
	Average Waiting Time in Each Workstation's Queue	5*, 9**
	Total Number of Features	**28*, 40****

* Flow shop model.

** Hybrid flow shop model.

A flow shop with five machines and a hybrid flow shop with nine workstations and multiple identical machines in each workstation, are built to imitate the production processes of a highly dynamic job shop, shown in Fig. 1 and 2 respectively. The capacity of each machine buffer is assumed to be infinite. The order arrival times are drawn from an exponential distribution and the shop utilisation rate of 85% is obtained by adjusting the inter arrival time of orders.

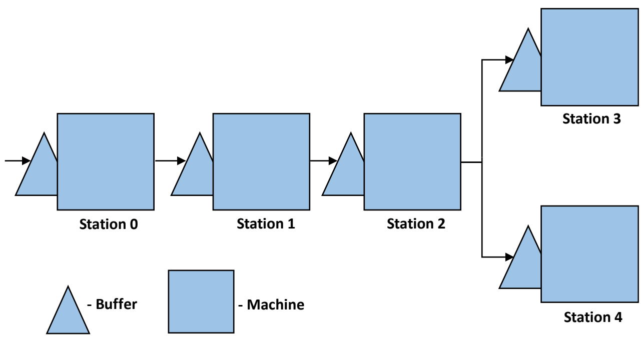

Fig. 1. Configuration of Flow Shop Model.

The characteristics of each order are generated on arrival and jobs are released to the system without delay. Several order routings are permitted representing various job types the hypothetical system manufactures. An individual job may be processed no more than once on any machine unless rework is required, the probability of which is kept constant at 4%. The batch size of an arriving order is drawn from a uniform distribution in the range [1,8]. The operation times are drawn from a continuous normal distribution, the mean and variance of which are known for a given job type. The machine set-up times are included in the operation times and transportation times are not considered. Machine failure and subsequent maintenance is considered. The occurrence of the former is drawn from an exponential distribution while the machine repair times are drawn from a continuous normal distribution, the mean and variance of which are known.

Two experiments were conducted. The first experiment was performed using a batch flow model framework. In this case, batches are not split and flow as a group between workstations. This means that all units in the batch must wait until all work is finished at a workstation before they can enter the queue of the next. The framework was then adapted for the second experiment. In this standard flow framework, jobs in the same order flow independently of one another. In this case, the first job of an order to be activated would record the status of the shop on arrival and the final job to be completed would record the order flowtime. Three dispatching rules (FCFS, SPT, EDD) are used for controlling the sequencing of jobs. The shop utilisation level is maintained at 85% representing a relatively heavy work load in the shop.

For each experiment conducted a simulation is run to generate sufficient data examples for training, validation and testing. A dataset containing 10,000 transactions is collected after a warm up period of 1000 orders to allow the system to reach a steady state. The data collected is then split into training, validation and test data. The prediction performance of the trained algorithms is first evaluated on an initial test dataset containing 2000 transactions. A final round of simulation experiments is carried out to collect data using a different seed
to allow the trained algorithms' prediction performance to also be evaluated on an unseen, external dataset containing 10,000 transactions.

4. Experimental results and analysis

In the experiments, the prediction performance of the five ML methods is compared with those of the two dynamic DDA rules (DTWK and DPPW). Mean Absolute Lateness (MAL), Root Mean Squared Lateness (RMSL) and Fill Rate (FR) are chosen as the evaluation criteria. MAL and RMSL are defined respectively by equation (3) and (4), where n is the total number of orders, L_i is the lateness of order i defined as $L_i = c_i - d_i$, c_i refers to the completion time or actual flowtime of order i and d_i denotes the estimated due date of order i. The FR is defined as: $FR = (n_E/n) \times 100$, where n_E is the number of early orders completed. The FR is calculated to give an

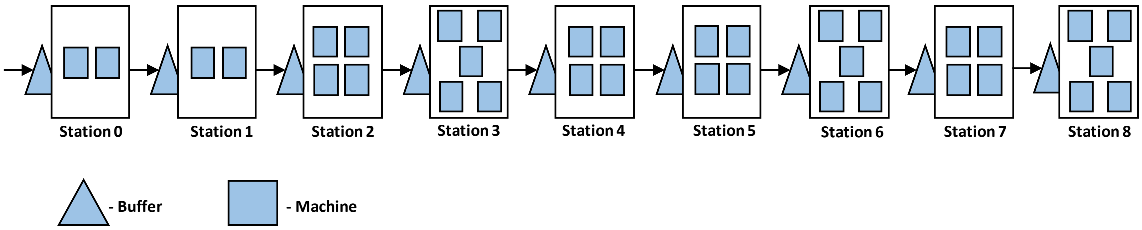

Fig. 2. Configuration hybrid flow shop model.

indication of the skew of the error prediction. A fill rate above 50% indicates that the method mostly over-estimated the due-date. This can be beneficial to a manufacturer as it leads to higher customer satisfaction due to orders being delivered on time however, there are also costs associated with orders being completed too early. A fill rate below 50% means that the method tends to under-estimate the due-date. This can lead to high levels of tardiness, which will ultimately lead to reduced customer satisfaction through missed deadlines.

$$MAL = \sum_{i=1}^{n} \frac{|L_i|}{n} \tag{3}$$

$$RMSL = \sqrt{\sum_{i=1}^{n} L_i^2/n} \tag{4}$$

The prediction results of the various DDA models are shown in Tables 2(a) and 2(b). In the tables, each value represents the average value of 10,000 test data examples. The best prediction values are shown in bold.

The experimental results show that the ML methods perform better than the conventional dynamic DDA rules. In general, the h2o ANN method tended to have the highest MAL and RMSL values amongst the five ML algorithms however, this method demonstrated greater stability than the neuralnet ANN method when a more complex hybrid structure was utilised. The potential benefits to management utilising this algorithm is due to the computational speed and efficiency that this method provides. The performance of this algorithm can be improved further with more data available to train the model on unseen cases. The neuralnet method provided good FR performance and provided the best results in terms of MAL and RMSL on the simpler flow shop structures however, its FR performance became less consistent with increased complexity. The stability of the algorithm is seen to fluctuate greatly as the variance of its predictive performance increases as indicated by its high RMSL values in the hybrid flow shop structure, notably for the batch flow experiment. This instability is caused by new levels in the shop conditions which the algorithm has not seen previously. This model is less efficient for training and is computationally expensive. Although it provided poor performance on more complex structures, this method obtained very high predictive accuracy for simpler manufacturing structures, demonstrating consistency in performance across all three dispatch policies.

The XGBoost method showed good consistency across all experiments with strong performance in FR values. This method is extremely efficient for training the algorithms and presents a viable option for production management to utilise in terms of ease and computational efficiency.

The Cubist method provides the second-best performance to the Random Forest method when initially tested on the datasets used for training and provides the best overall performance when tested on the external datasets. This method demonstrates good consistency in results for MAL, RMSL and RF performance measures. An advantage of this method is that the rules provided from the trained algorithms give further insights into the manufacturing operations and importance of shop conditions. In the case where predictions are made based on the datasets used for training the models, the random forest method consistently provides the best performance in terms of MAL and MSL for both standard and batch flow conditions, while the best performance in terms of fill rate tends to be distributed amongst the four other methods. The results are quite accurate especially in the simpler flow shop structures. The model demonstrates good stability when tested on the external datasets, which indicates that the model is not overfitted to the training data.

In general, the prediction accuracy for the standard flow experiment was greater. This occurs since jobs within the same order can be processed in parallel or may continue to the next workstation once they have been processed, which leads to less variance in flowtime due to batch processing. Batch flow is most applicable to environments where there is some transport restriction between stations. The datamining Cubist method was originally proposed for such an environment. The random forest method provided the best results from the training datasets. When tested on external datasets, neuralnet was observed to provide the best predictive accuracy for the simpler manufacturing structures, while the Cubist and Random Forest methods provided the best performance for more complex manufacturing structures. The results from the experiments demonstrated that factors, such as the flow of the manufacturing line, the processing times at machines and the contents of orders all have an effect on the flowtime prediction. The performance of the method predictions is improved by using information intensive methods, factoring in job and shop related conditions rather than using simpler methods. The bottleneck position was shown to impact the variance of the predictions with manufacturing lines where the bottleneck location is further from the start of the production process providing more consistent prediction performance with lower variation. This can be explained since the prediction is made regarding the status of the shop at the time of arrival. It is assumed that most of the work in the system resides in the bottleneck queue and there may be a significant change to the status of the system when the job eventually enters the bottleneck's queue. The performance of the models suffered more when longer processing times and more complex manufacturing structures were introduced, especially for the SPT case.

Table 2 (a). Flow shop prediction results.

Batch Flow	Dispatch Policy								
	FCFS			SPT			EDD		
	MAL	RMSL	FR	MAL	RMSL	FR	MAL	RMSL	FR
H2O	12.47	15.09	72.4%	27.42	13.34	73.41%	16.72	20.21	72.97%
nn	**4.3**	**5.47**	54.86%	**15.26**	**21.92**	**50.11%**	**9.22**	**11.59**	73.12%
XGBoost	6.85	8.82	**53.51%**	21.81	30.07	61.59%	10.81	14.13	58.55%
Cubist	6.44	8.37	38.83%	19.1	27.61	57.84%	9.82	12.67	55.86%
RF	6.57	8.45	54.68%	21.54	29.37	68.85%	10.1	13.13	**55.31%**
DTWK	53.76	69.48	5.03%	53.26	72.33	6.15%	47.65	62.97	5.25%
DPPW	40.66	58.19	19.34%	42.3	62.29	23.83%	34.24	50.72	18.76%

Standard Flow	Dispatch Policy								
	FCFS			SPT			EDD		
	MAL	RMSL	FR	MAL	RMSL	FR	MAL	RMSL	FR
H2O	8.23	10.53	67.41%	18.93	26.9	**46.24%**	13.51	16.43	78.79%
nn	**3.57**	**4.58**	62.46%	**12.7**	**17.86**	60.23%	**5.68**	**7.52**	59.14%
XGBoost	4.55	6.23	53.73%	19.27	25.75	65.37%	9.11	12.35	46.44%
Cubist	4.21	5.77	**52.38%**	17.91	24.59	62.05%	8.38	11.31	53.2%
RF	4.36	35.74	54.74%	19.22	25.3	68.7%	8.67	11.75	57.14%
DTWK	34.45	48.96	30.78%	40.67	57.5	26.6%	26.97	42.28	36.59%
DPPW	29.2	42.62	42.92%	33.76	48.77	33.89%	22.05	35.74	**51.38%**

Table 2(b). Hybrid flow shop prediction results.

Batch Flow	Dispatch Policy								
	FCFS			SPT			EDD		
	MAL	RMSL	FR	MAL	RMSL	FR	MAL	RMSL	FR
H2O	14.84	18.33	69.05%	22.5	29.51	**49.63%**	18.25	23.84	**53.07%**
nn	30.38	50.72	58.15%	47.16	64.09	40.94%	103.32	142.51	79.89%
XGBoost	18.26	22.02	80.63%	22.14	28.45	72.42%	16.33	20.93	61.67%
Cubist	**11.32**	**14.47**	**55.76%**	**15.38**	**22.16**	52.16%	**13.7**	**17.64**	61.35%
RF	11.76	14.91	58.51%	20.26	26.07	75.24%	14.05	17.81	61.18%
DTWK	69.28	193.83	38.25%	69.11	193.58	39.9%	70.86	195.41	39.43%
DPPW	101.98	198.32	87.78%	103	198.55	88.12%	104.16	199.83	87.79%

Standard Flow	Dispatch Policy								
	FCFS			SPT			EDD		
	MAL	RMSL	FR	MAL	RMSL	FR	MAL	RMSL	FR
H2O	4.48	5.65	66.11%	13.84	20.81	30.36%	7.38	9.31	66.12%
nn	5.61	8.3	22.44%	16.69	23.63	**39.68%**	16.91	22.62	42.03%
XGBoost	3.99	4.99	64.42%	14.58	19.27	74.96%	6.34	8.67	**55.55%**
Cubist	3.53	4.62	**47.24%**	**10.58**	**15.83**	62.58%	**5.48**	**7.42**	41.74%
RF	**3.51**	**4.5**	53.08%	12.56	17.06	74.83%	5.9	7.81	57.86%
DTWK	76.47	103.95	85.01%	71.15	95.01	84.41%	76.59	102.78	91.37%
DPPW	91.04	117.3	88.61%	83.1	106.43	88.43%	92.36	116.57	94.04%

However, it was observed that better results were obtained in the batch flow experiment with longer processing times.

The stability of the neural net-based ANN model's performance was drastically reduced as the shop complexity increased. In general, for standard flow shops the best predictive performance in terms of MAL and RMSL was achieved under FCFS dispatch policy with SPT providing the worst performance in terms of difficulty in accurately estimating flowtime. The best FR results were achieved when EDD was utilised which is to be expected since the flow of jobs through the system is dictated based on initial due-date estimates of each order.

5. Conclusions and future research

In this paper, a modelling framework has been developed for the rapid analysis of flow shop models. This framework was then modified to collect data for training purposes, and a series of discrete event simulation experiments were then designed to investigate the use of five ML methods for flowtime estimation in a dynamic job shop model. These methods were tested on various shop layouts, under three different dispatching rules.

The performance of the methods indicate that production managers can utilise the available information to predict flowtime and assign due-dates more accurately and with greater relative ease using open source software and machine learning technologies. The significant and rapid improvements in data ingestion and integration technologies make ML a powerful tool in manufacturing processes for production management and can be utilised to gain beneficial insights into the behaviour of manufacturing systems, improving the quality of decisions. While most of the earlier studies have ignored more complex shop behaviour, this study attempts to explore the application of ML techniques for flowtime prediction when factors such as rework, machine downtime, machine repair time and machine set-up times are considered for a combination of flow shop and hybrid flow shop systems.

Key challenges include the adoption of advanced manufacturing technologies, agile and flexible enterprise capabilities and supply chains, and close the gap between industry and academia to adopt new technologies. Further work in this area could be performed to (1) test these methods on a range of shop utilisation levels with larger datasets, possibly achieving a close cooperation between industry and research to carry out further analysis on real-world systems, (2) evaluate their performance on a wider supply chain, (3) create a more refined attribute selection for training the models taking into account common flow shop features such as bottleneck related conditions and the use of order review and release policy (ORR) for more controlled release of orders to the system.

Modern manufacturing systems, featuring advanced, high-added value technologies, such as additive manufacturing, may benefit a great deal from developments where order placement and execution as well as system performance analysis may be achieved without building complex simulation models.

Acknowledgements

This work has been partially supported by a research grant from Science Foundation Ireland (SFI) under Grant Number 16/RC/3872 and is co-funded under the European Regional Development Fund.

References

[1] Walker J. Machine Learning in Manufacturing – Present and Future Use-Cases: Emerj; 2018 [cited 2018 24/2/2018]. Available from: https://emerj.com/ai-sector-overviews/machine-learning-in-manufacturing/.
[2] Yang DH, Hu L, Qian Y. Due Date Assignment in a Dynamic Job Shop with the Orthogonal Kernel Least Squares Algorithm. IOP Conf Ser-Mat Sci. 2017;212(1):012022.
[3] Lechevalier D, Narayanan A, Rachuri S, editors. Towards a domain-specific framework for predictive analytics in manufacturing. BigData Conference; 2014.
[4] Iida T. Benefits of leadtime information and of its combination with demand forecast information. Int J Prod Econ. 2015;163:146-56.
[5] Ragatz GL, Mabert VA. A simulation analysis of due date assignment rules. J Oper Manag. 1984;5(1):27-39.
[6] Cheng T, Jiang J. Job shop scheduling for missed due-date performance. Comput Ind Eng. 1998;34(2):297-307.
[7] Little JD. A proof for the queuing formula: L= λ W. Oper Res. 1961;9(3):383-7.
[8] Patil RJ. Using ensemble and metaheuristics learning principles with artificial neural networks to improve due date prediction performance. Int J Prod Res. 2008;46(21):6009-27.
[9] Philipoom PR, Rees LP, Wiegmann L. Using Neural Networks to Determine Internally-Set Due-Date Assignments for Shop Scheduling. Decision Sci. 1994;25(5-6):825-51.
[10] Sha D, Hsu S. Due-date assignment in wafer fabrication using artificial neural networks. Int J Adv Manuf Technol. 2004;23(9-10):768-75.
[11] Silva C, Ribeiro V, Coelho P, Magalhães V, Neto P. Job shop flow time prediction using neural networks. Procedia Manuf. 2017;11:1767-73.
[12] Öztürk A, Kayalıgil S, Özdemirel NE. Manufacturing lead time estimation using data mining. Eur J Oper Res. 2006;173(2):683-700.
[13] Sha D, Storch R, Liu C-H. Development of a regression-based method with case-based tuning to solve the due date assignment problem. Int J Prod Res. 2007;45(1):65-82.
[14] Sha D, Liu C-H. Using data mining for due date assignment in a dynamic job shop environment. Int J Adv Manuf Technol. 2005;25(11-12):1164-74.
[15] Nguyen S, Zhang M, Johnston M, Tan KC. Genetic programming for evolving due-date assignment models in job shop environments. Evol Comput. 2014;22(1):105-38.
[16] Nguyen S, Zhang M, Johnston M, Tan KC. Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming. IEEE T Evolut Comput. 2014;18(2):193-208.
[17] Chryssolouris G, Dicke K, Lee M. On the Resources Allocation Problem, Int. J. Prod. Res., 1992;30(12): 2773-2795.
[18] Chryssolouris G, Pierce J, Dicke K. An Approach for Allocating Manufacturing Resources to Production Tasks. J. Manuf. Syst., 1991;19(5): 368-382.
[19] Chryssolouris G, Papakostas N, Mourtzis D. A Decision Making Approach for Nesting Scheduling: A Textile Case, Int. J. Prod. Res., 2000;28(17): 4555-4564.
[20] Chryssolouris G, Mourtzis D, Geronimaki M. An approach to planning of industry: A case study for a refrigerators producing facility, CIRP J. Manuf. Syst., 2003;32(6): 499-506.
[21] R Core Team. R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, 2018, https://www.R-project.org.
[22] Page B, Kreutzer W. The Java Simulation Handbook: Simulating Discrete Event Systems, 2005, Shaker Verlag.