Due Date Assignment in a Dynamic Job Shop with the Orthogonal Kernel Least Squares Algorithm

D.H. Yang, L. Hu, Y. Qian
School of Business Administration, Shenzhen Institute of Information Technology, Shenzhen, China
yangdh@sziit.edu.cn; hulingedu@126.com; 4091172@qq.com

Abstract. Meeting due dates is a key goal in the manufacturing industries. This paper proposes a method for due date assignment (DDA) by using the Orthogonal Kernel Least Squares Algorithm (OKLSA). A simulation model is built to imitate the production process of a highly dynamic job shop. Several factors describing job characteristics and system state are extracted as attributes to predict job flow-times. A number of experiments under conditions of varying dispatching rules and 90% shop utilization level have been carried out to evaluate the effectiveness of OKLSA applied for DDA. The prediction performance of OKLSA is compared with those of five conventional DDA models and back-propagation neural network (BPNN). The experimental results indicate that OKLSA is statistically superior to other DDA models in terms of mean absolute lateness and root mean squares lateness in most cases. The only exception occurs when the shortest processing time rule is used for dispatching jobs, the difference between OKLSA and BPNN is not statistically significant.

Keywords: Due Date Assignment; Job Shop; Orthogonal Kernel Least Squares Algorithm

1. Introduction
In order to remain competitive in today’s global markets, modern manufacturing systems often work in dynamic environments. Aside from providing high quality products, meeting due dates is another crucial requirement of production control. The “due date” is the specified date when production of a particular order is expected to be complete [1]. No matter imposed externally by the customers or set internally by the manufacturer, the due date related performance has a great impact upon the economic benefit of the production [2]. A good due date assignment (DDA) rule can perform timely delivery and reduce inventory costs, thereby enhancing the customer satisfaction and competitive advantage [3]. A typical DDA procedure is to estimate the flow-time of an arriving job and then adding that to its arrival time. It is very simple to estimate the job flow-time and set the due date in a purely deterministic system. However, estimating the accurate flow-time is not an easily-solved problem in a dynamic environment, due to the stochastic nature of arriving jobs and the uncertainty of processing sequences caused by some dispatching rules.

Owing to the critical importance of due dates, the DDA problem in a dynamic, multi-machine manufacturing system has been studied for a long time. During the first years, studies were focused on simple, regression based approaches. Some conventional rules, e.g. TWK, NOP, TWK+NOP, JIQ, WIQ, WEEKS, JIS and RMR, were proposed by different researchers and a thorough comparison was conducted in [2]. These rules employ different kinds of job characteristics and shop state information.
to predict job completion times. The comparison results demonstrate that information related to the machine congestion along a job’s routing is more useful than general shop conditions.

In general, the aforementioned conventional DDA rules rely on linear or low-order nonlinear regression methods. They lack the ability to reveal the complex nonlinear relationship between actual job flow-times and relevant information concerning job characteristics and system state. To overcome the shortcoming of these rules, a number of researchers subsequently applied artificial intelligence approaches to assign due dates in job shops or wafer fabrication. Neural Networks (NNs) were first employed to specify the internally-set due dates in dynamic job shops [4]. After that, NNs continued to show effectiveness and robustness when they were combined with other techniques in varying production environments [5, 6]. Except for NNs, other artificial intelligence approaches were also incorporated to improve the DDA performance, such as Self-Organizing Map and fuzzy rule [7].

In this paper, we propose a DDA method by using the Orthogonal Kernel Least Squares Algorithm (OKLSA) [8, 9]. OKLSA aims at finding an approximate solution for the kernel least squares problem and obtains a regression model with a sparse structure. A discrete-event simulation model is first built to imitate the production process of a highly dynamic job shop. A series of simulation runs are then conducted with varying dispatching rules (FIFO, SPT, EDD) to collect data set for training, validating and testing different DDA models. Some numerical experiments are carried out to evaluate the DDA performance of OKLSA by comparing its prediction result with five conventional rules and Back-Propagation Neural Network (BPNN) in terms of Mean Absolute Lateness (MAL) and Root Mean Squares Lateness (RMSL) criteria. The experimental results indicate that OKLSA statistically outperforms other DDA models in most cases. The only exception occurs when SPT rule is used for dispatching jobs, the difference between OKLSA and BPNN is not statistically significant.

The remainder of the paper is organized as follows. Section 2 reviews the basic theory of OKLSA. Section 3 describes the OKLSA-based methodology for due date assignment. Furthermore, section 4 introduces the experimental design and data generation. Section 5 shows the experimental results and gives a brief analysis. The conclusions and suggestions for future work are presented in Section 6.

2. Orthogonal Kernel Least Squares Algorithm[8, 9]

2.1. Kernel Least Squares Algorithm

Assume to be given a training data set \( \{(x_i, y_i)\}_{i=1}^{n} \) with input vectors \( x_i \in \mathbb{R}^n \) and associated target outputs \( y_i \in \mathbb{R} \). Kernel Least Squares Algorithm (KLSA) consists of finding a function:

\[
 f(x) = w^T \varphi(x) + b ,
\]

where the mapping \( \varphi(\cdot) \) is implemented via a kernel that satisfies the Mercer’s condition such that \( K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) \). The output \( y_i \) can be expressed as:

\[
 y_i = f(x_i) + \zeta_i = w^T \varphi(x_i) + b + \zeta_i ,
\]

where \( \zeta_i \) is the training error. In accordance with the Representer Theorem [10], the coefficient vector \( w \) of the optimal solution of (1) must lie in the span of all training data points in the feature space, in the form of

\[
 w = \sum_{i=1}^{n} \alpha_i \varphi(x_i)
\]

where \( \alpha_i (i = 1, \ldots, n) \) are the coefficients. Substituting (3) into (2) and omitting the threshold yields:

\[
 y_i = f(x_i) + \zeta_i = \sum_{i=1}^{n} \alpha_i K(x_i, x) + \zeta_i .
\]

Expression (4) can be reformulated with the concise form

\[
 y = K\alpha + \Xi ,
\]
where \( \mathbf{y} = [y_1, \ldots, y_n]^T \), \( \mathbf{K} = [\mathbf{k}_1, \ldots, \mathbf{k}_n] = \begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_n) \\ \vdots & \ddots & \vdots \\ K(x_n, x_1) & \cdots & K(x_n, x_n) \end{bmatrix} \), \( \mathbf{a} = [\alpha_1, \ldots, \alpha_n]^T \), and \( \mathbf{\Xi} = [\xi_1, \ldots, \xi_n]^T \).

The regression model can be obtained by minimizing the sum of squared residual errors as:

\[
\min \| \mathbf{K} \mathbf{a} - \mathbf{y} \|^2.
\]  

(6)

2.2. Orthogonal Kernel Least Squares Algorithm

The computational complexity and memory requirement for computing the solution of (6) is prohibitive for very large datasets. In addition, the "dense" structure of the yielded model is easy to cause the overfitting problem. An alternative approach to overcome these difficulties would be to enforce regularization via "sparsity", with which the coefficient vector \( \mathbf{w} \) in (3) can be approximated by

\[
\mathbf{w} \approx \sum_{j=1}^{m} \alpha_j \varphi(\mathbf{x}_j^*)
\]  

(7)

where \( m \ (< n) \) is a positive integer and \( \mathbf{x}_j^* \in \{ \mathbf{x}_i | i = 1, \ldots, n \} \). Problem (6) hence yields an approximation problem as:

\[
\min \| \mathbf{K}^* \mathbf{a}^* - \mathbf{y} \|^2,
\]  

(8)

where \( \mathbf{a}^* = [\alpha_1, \ldots, \alpha_m]^T \), \( \mathbf{y} = [y_1, \ldots, y_m]^T \), and \( \mathbf{K}^* = \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1^*) & \cdots & K(\mathbf{x}_1, \mathbf{x}_m^*) \\ \vdots & \ddots & \vdots \\ K(\mathbf{x}_m, \mathbf{x}_1^*) & \cdots & K(\mathbf{x}_m, \mathbf{x}_m^*) \end{bmatrix} \) being a rectangular matrix formed by choosing a subset of \( m \) columns from the original kernel matrix \( \mathbf{K} \). The final function is then expressed as \( f(\mathbf{x}) = \sum_{j=1}^{m} \alpha_j K(\mathbf{x}_j^*, \mathbf{x}) \). When \( m \) is a small number, the obtained regression model has a sparse structure. Since it is computationally demanding to obtain the optimal solution of (8), research efforts have been concentrated on developing methods to obtain sub-optimal solutions.

To get a sub-optimal solution of (8), OKLSA attempts to select a subset of the most significant terms for the regression model in a sequential forward manner. It calculates the individual contribution of each kernel term \( \varphi(\mathbf{x}_i) \) to error minimization by transforming the set of \( \mathbf{k}_i \) \( (i = 1, \ldots, n) \) into a set of orthogonal vectors. Hence, the matrix \( \mathbf{K} \) is decomposed into the product of two matrices represented as \( \mathbf{K} = \mathbf{UB} \), where \( \mathbf{B} \in \mathbb{R}^{m \times m} \) is an upper-triangular matrix with 1’s on the diagonal and \( \mathbf{U} \in \mathbb{R}^{m \times m} \) is an matrix with orthogonal columns such that \( \mathbf{U}^T \mathbf{U} = \mathbf{T} \), where \( \mathbf{T} \) is diagonal and the diagonal elements are calculated by \( t_i = \mathbf{u}_i^T \mathbf{u}_i \). Therefore, the space spanned by the set of orthogonal vectors \( \mathbf{u}_i \) is the same as the space that is spanned by the set of \( \mathbf{k}_i \). Equation (5) can be rewritten as:

\[
\mathbf{y} = (\mathbf{KU}^+)(\mathbf{Ua}) + \mathbf{\Xi} = \mathbf{Tg} + \mathbf{\Xi},
\]  

(9)

When minimizing the total squared errors \( \mathbf{\Xi}^T \mathbf{\Xi} \), its solution is given by

\[
\mathbf{g} = \mathbf{T}^{-1} \mathbf{U}^T \mathbf{y},
\]  

(10)

or \( g_i = \mathbf{u}_j^T \mathbf{y} / (\mathbf{u}_j^T \mathbf{u}_j) \) \( (i = 1, \ldots, n) \). The values of the vectors \( \mathbf{g} \) and \( \mathbf{a} \) satisfy \( \mathbf{Ba} = \mathbf{g} \). Knowing \( \mathbf{B} \) and \( \mathbf{g} \), \( \mathbf{a} \) can be determined through back substitution. Due to the fact that \( \mathbf{u}_i \) and \( \mathbf{u}_j \) are orthogonal for \( i \neq j \), the total squares errors can be represented as:

\[
\mathbf{\Xi}^T \mathbf{\Xi} = \mathbf{y}^T \mathbf{y} - \sum_{i=1}^{n} g_i^2 \mathbf{u}_i^T \mathbf{u}_i
\]  

(11)
Equation (11) implies that the error reduction due to $\mathbf{u}_i$ is defined as:

$$ [err]_i = g_{ii}' \mathbf{u}_i (y^T y). $$  \hfill (12)

OKLSA selects the kernel terms one by one according to their contributions to residual error minimization. According to (12), it is convenient to select a couple of most significant kernel terms, which leads to the greatest reduction in the residual errors. When applying OKLSA to solving (8), the selection of a subset of significant kernel terms is equivalent to the selection of a subset of columns from the kernel matrix.

3. **OKLSA-based Methodology for Due Date Assignment**

The due date of a given job should be equal to the sum of the job’s arrival time and flow-time in the system. This can be represented by the equation: $D_i = A_i + F_i$, where $D_i$ represents the due date of job $i$, $A_i$ is its arrival time, and $F_i$ is its flow-time. The typical DDA procedure therefore consists of estimating the flow-time of the arriving job and then adding the estimated flow-time to the job’s arrival time accordingly. The methodology for DDA with OKLSA can be described as follows:

1. A discrete-event simulation model is first tailored according to the system configuration and the dispatching rule used for controlling the job sequence.
2. For each particular set of information inputs, a simulation is run for a lengthy period to generate a set of data. The whole dataset is segmented into three subsets: the training set, the validation set and the test set.
3. A DDA model is subsequently constructed with OKLSA based on the training data.
4. The data in the validation set is used to determine the optimal parameters in OKLSA and the effectiveness of the obtained DDA model is evaluated by the data in the test set.

4. **Experimental Design**

To study the prediction performance of various DDA models, we build a discrete-event simulation model with C++ language to imitate the production process of a highly dynamic job shop. The shop has nine machines and the size of each machine buffer is assumed to be unlimited. The job inter-arrival times are drawn from an exponential distribution. The desired shop utilization rate is yielded by adjusting the job arrival rate. The characteristics of each job are generated on arrival. Jobs entering the shop are randomly routed with no consecutive operations on the same machine permitted. The number of operations per job is drawn from a uniform distribution in the range from 1 to 9. The operation times are drawn from an exponential distribution with a mean of 1. The set-up times are included in the operation times and the transportation times are negligible. It is assumed that there are no mechanical failures or required maintenance on all the machines during the simulation horizon.

To evaluate the relative effectiveness and robustness of DDA models under varying dispatching rules, three different dispatching rules (FIFO, SPT and EDD) are used for controlling the job sequence. The shop utilization level is kept at 90%, which indicates that the work load in the shop is relatively heavy. For a particular system situation, a simulation is run to generate sufficient data examples for training, validation and test. Each data example is an “attributes – target value” pair. The “attributes” are those variables describing job characteristics and system state information. The “target value” refers to the job flow-time. Seven independent attributes are selected as [4] and listed in Table 1.

Once the simulation begins, the first 500 jobs are treated as a warm-up period. When the system reaches the steady state, data recordation begins. At the instant job $i$ arrives at the system, the aforementioned seven attributes are recorded and form a vector $\mathbf{x}_i$. When job $i$ is completed and exits the system, its flow-time through the system is recorded as a scalar $y_i$. Hence, each data example is notated in an “attributes – target value” pair represented as $(\mathbf{x}_i, y_i)$. To reduce the possibility of serial correlation and guaranteeing statistical independence, one data example is selected from every twenty jobs. A total of 13,000 data examples are collected. All data examples are randomly permuted and
independently divided into three sets. The first set includes 1,000 data examples as training data, which are used to estimate the regression parameters of the five conventional rules as well as to train BPNN and OKLSA. The second set involving 2,000 data examples is the validation set to determine the optimal parameters used in BPNN and OKLSA. The remaining 10,000 data examples comprise the third set used for evaluating the prediction performance of various DDA models. Moreover, to avoid the numerical problems in OKLSA and BPNN, each attribute is linearly scaled to the range [-1,+1].

Table 1. The selected attributes describing job characteristics and system state information.

| Index | Attributes description |
|-------|------------------------|
| 1     | Number of operations required by job i |
| 2     | Total processing time for job i |
| 3     | Number of operations that must be done by the machine on job i’s path to complete all jobs presently in the shop |
| 4     | Sum of jobs presently in queues on job i’s routing |
| 5     | Maximum processing time job i requires on any machine |
| 6     | Number of operations required to empty the shop of its current workload |
| 7     | Total processing time of jobs in queues on job i’s routing |

5. Experimental Results and Analysis

In the experiments, the prediction performance of OKLSA is compared with those of the aforementioned five conventional rules (i.e. TWK, NOP, TWK+NOP, JIQ and WIQ) and BPNN. Mean Absolute Lateness (MAL) and Root Mean Squares Lateness (RMSL) are chosen as the criteria [11, 12]. MAL and RMSL are defined respectively as: \[ MAL = \frac{1}{n} \sum_{i=1}^{n} L_i, \] \[ RMSL = \sqrt{\frac{1}{n} \sum_{i=1}^{n} L_i^2}, \] where \( L_i = AF_i - PF_i, \) \( n \) is the total number of jobs, \( L_i \) is the lateness of job \( i \), \( AF_i \) refers to the actual flow-time of job \( i \), and \( PF_i \) denotes the predicted flow-time of job \( i \). The coefficients of the five conventional rules are calculated by minimizing the variance between predicted flow-times and actual flow-times among the training data. BPNN employs a common three-layer neural network with the 7-10-1 structure and is trained by the back-propagation algorithm. Sigmoid transfer function is used in the hidden-layer, and linear function is used in the output-layer. The RBF kernel \( K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \) is adopted in OKLSA, where \( \gamma \) is selected in the range \( \{2^2, 2^4, ..., 2^2 \} \). The number of kernel terms used in OKLSA is selected in the range \( m \in \{4, 6, 8, 10, 12, 14, 16, 20, 22, 24, 26, 28, 30, 32, 35\} \). The optimal set of parameters (of OKLSA or BPNN) is determined when the prediction error on the validation set is minimized. This set of parameters is then used to train a DDA model for evaluating the prediction error on the test set. The prediction results of various DDA models are shown in Tables 2(a) and 2(b). In the tables that list MAL or RMSL, each value represents the average value of 10,000 test data examples. The best prediction values are written in the bold and italic form.

Table 2(a). Prediction results in terms of MAL.

|       | FIFO | SPT | EDD |
|-------|------|-----|-----|
| TWK   | 22.95| 12.68| 14.21|
| NOP   | 19.02| 14.52| 17.57|
| TWK+NOP| 18.96| 12.75| 13.92|
| JIQ   | 9.40 | 12.65| 9.93 |
| WIQ   | 8.37 | 12.72| 10.03|
| BPNN  | 8.33 | **11.15**| 7.70 |
| OKLSA | **8.05**| 11.22| **7.50**|

Table 2(b). Prediction results in terms of RMSL.

|       | FIFO | SPT | EDD |
|-------|------|-----|-----|
| TWK   | 31.69| 32.97| 18.98|
| NOP   | 25.73| 36.34| 22.80|
| TWK+NOP| 25.62| 32.32| 18.14|
| JIQ   | 13.76| 32.96| 12.87|
| WIQ   | 12.80| 32.97| 13.03|
| BPNN  | 12.48| 30.69| 10.44|
| OKLSA | **12.07**| **30.42**| **9.95**|
Just like the prior research [4], the statistical test (the paired t-test) is carried out to check whether OKLSA is significantly superior to the other methods. The 10,000 test data examples are grouped into ten batches and paired t-tests are then conducted using the batch means to test the hypotheses. Since the number of test data examples is large, the MAL or RMSL of each batch are regarded as being approximately uncorrelated and normally distributed according to the central limit theorem. This condition thereby meets the assumptions of the paired t-test. One-tailed hypotheses are formulated that the MAL or RMSL for each of the conventional rules or BPNN do not exceed that of OKLSA. The hypotheses are hoped to be rejected. In particular: \( H_0 : \mu_0 \leq 0; \ H_1 : \mu_0 > 0 \), where \( D \) denotes the MAL (or RMSL) difference between the conventional rules (or BPNN) and OKLSA. For all hypotheses tests, the level of significance is set at \( \alpha = 0.05 \). The Bonferroni’s inequality is used to control the experiment-wise error rate of the six multiple comparisons. Therefore the \( p \)-value must be less than \( 0.05 / 6 = 0.0083 \) for comparisons to be statistically significant. Results of the statistical tests are listed in the form of \( t \)-values and \( p \)-values for the MAL or RMSL criterion in Tables 3(a) and 3(b).

| Table 3(a). Statistical test \( t \)-values and \( p \)-values (MAL). |
|-------------------|-------------------|-------------------|
| FIFO              | SPT               | EDD               |
| TWK               | 29.0562           | 0.0001            | 13.5121           | 0.0001            | 16.7889           | 0.0001            |
| NOP               | 31.8841           | 0.0001            | 15.6180           | 0.0001            | 32.6923           | 0.0001            |
| OKLSA versus NOP  | 30.6449           | 0.0001            | 17.8034           | 0.0001            | 16.1800           | 0.0001            |
| JIQ               | 23.5956           | 0.0001            | 13.1993           | 0.0001            | 16.8179           | 0.0001            |
| WIQ               | 6.0939            | 0.0001            | 14.3742           | 0.0001            | 16.4340           | 0.0001            |
| BPNN              | 9.3715            | 0.0001            | -0.7568           | 0.7675            | 6.6434            | 0.0001            |

| Table 3(b). Statistical test \( t \)-values and \( p \)-values (RMSL). |
|-------------------|-------------------|-------------------|
| FIFO              | SPT               | EDD               |
| TWK               | 23.6865           | 0.0001            | 4.4193            | 0.0008            | 14.4895           | 0.0001            |
| NOP               | 19.7061           | 0.0001            | 8.2708            | 0.0001            | 26.3573           | 0.0001            |
| OKLSA versus NOP  | 19.0961           | 0.0001            | 4.1633            | 0.0012            | 13.8695           | 0.0001            |
| JIQ               | 20.2452           | 0.0001            | 4.4397            | 0.0008            | 12.8460           | 0.0001            |
| WIQ               | 6.8000            | 0.0001            | 4.3797            | 0.0008            | 12.3404           | 0.0001            |
| BPNN              | 7.1634            | 0.0001            | 0.9039            | 0.1948            | 6.7368            | 0.0001            |

The hypotheses test results show that the calculated \( p \)-values are less than 0.0083 in most cases. It is clearly indicated that the null hypothesis is rejected for most hypotheses at an alpha-level of 0.05 and OKLSA statistically outperforms the conventional rules and BPNN in most cases. OKLSA is able to predict job flow-times with higher accuracy than the five conventional rules in all cases. Due to the employment of kernel function, OKLSA is able to construct nonlinear models to reveal the complicated relationship that exists between the job flow-times and the influential factors. All the conventional DDA rules use linear regression methods to construct simple DDA models. Although they are easy to implement and comprehend, their performance is relatively poor mainly because such linear models are easy to cause the “underfitting” problem. Hence, it is reasonable that OKLSA is significantly superior to the conventional rules.

OKLSA also substantially surpasses BPNN, with the only exception when SPT being used for dispatching jobs within the system. The advantage of OKLSA over BPNN lies in two points: (1) the use of kernel function enables OKLSA to learn the training data in the high-dimensional feature space; and (2) OKLSA is able to construct models with a sparse structure, which makes OKLSA easily generalize the learned knowledge to the newly arrived data. In the case that SPT is employed, OKLSA shows a similar prediction accuracy to BPNN on the RMSL criterion but a little worse than BPNN on the MAL criterion, although the difference is not statistically significant. When the SPT rule is applied
to sequencing job operations, some jobs with very large lateness are likely to occur. When there are a large number of such very-late jobs in the data set, the prediction performance of OKLSA may be heavily influenced by them.

6. Conclusions and Future Research
This paper constructs a sparse nonlinear regression model to assign due date in a dynamic job shop by using the Orthogonal Kernel Least Square Algorithm. A discrete-event simulation model has been built and some simulation experiments have been carried out to evaluate the performance of OKLSA by comparing its prediction accuracy with other DDA models. The experimental results indicate that OKLSA statistically outperforms all the conventional DDA rules and BPNN with regard to both MAL and RMSL criteria in most cases. The only exception occurs when the SPT rule is used for dispatching jobs within the system. Influenced by very-late jobs, OKLSA shows similar performance to BPNN on the RMSL criterion but is slightly poorer than BPNN on the MAL criterion, although the difference is not statistically significant. Due to the desirable prediction performance achieved in most cases, OKLSA is therefore proven to be a very promising methodology for DDA in the dynamic job shops.

In this study, OKLSA has not shown the best performance when the SPT rule is used in the job shop. It is thus important to develop an effective strategy or technique in the future to enhance OKLSA's performance in the SPT case. Moreover, OKLSA suffers from expensive computational costs when the dataset is very large. One promising area for further research would be to derive more efficient training algorithms and apply them to solve DDA problems.

Acknowledgments
In this paper, the research was sponsored by the "The 1st-Class Excellent High Vocational College Project of Guangdong Province", "The 6th Teaching Research Project of Shenzhen Institute of Information Technology (Project No. 2016jgyb04) " and "The 13th Five-year Plan Education Project in Shenzhen (Project No. ybzz16009)".

References
[1] Eilon, S. and I.G. Chowdhury, Due dates in job shop scheduling. The International Journal of Production Research, 1976. 14(2): pp. 223-237.
[2] Ragatz, G.L. and V.A. Mabert, A simulation analysis of due date assignment rules. Journal of Operations Management, 1984. 5(1): pp. 27-39.
[3] Sha, D.Y. and C.H. Liu, Using Data Mining for Due Date Assignment in a Dynamic Job Shop Environment. International Journal of Advanced Manufacturing Technology, 2005. 25(11-12): pp. 1164-1174.
[4] Philipoom, P.R., L.P. Rees and L. Wiegmann, Using neural networks to determine internally-set due-date assignments for shop scheduling. Decision Sciences, 1994. 25(5): pp. 825-851.
[5] Hsu, S.Y. and D.Y. Sha, Due date assignment using artificial neural networks under different shop floor control strategies. International Journal of Production Research, 2004. 42(9): pp. 1727-1745.
[6] Patil, R.J., Using ensemble and metaheuristics learning principles with artificial neural networks to improve due date prediction performance. International Journal of Production Research, 2007. 46(21): pp. 6009-6027.
[7] Chang, P. and T.W. Liao, Combining SOM and fuzzy rule base for flow time prediction in semiconductor manufacturing factory. Applied Soft Computing, 2006. 6(2): pp. 198-206.
[8] Billings, S.A. and K.L. Lee, Nonlinear Fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm. Neural Networks, 2002. 15(2): pp. 263-270.
[9] Nair, P.B., A. Choudhury and A.J. Keane, Some greedy learning algorithms for sparse regression and classification with mercer kernels. The Journal of Machine Learning Research, 2002. 3: pp. 781-801.
[10] Kimeldorf, G. and G. Wahba, A Correspondence Between Bayesian Estimation on Stochastic Processes and Smoothing by Splines. Annals of Mathematical Statistics, 1970. 41(2): pp. 495-502.

[11] Vig, M.M. and K.J. Dooley, Dynamic rules for due-date assignment. The International Journal of Production Research, 1991. 29(7): pp. 1361-1377.

[12] Sabuncuoglu, I. and A. Comlekci, Operation-based flowtime estimation in a dynamic job shop. Omega, 2002. 30(6): pp. 423-442.