International Journal of Production Research
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tprs20

Development of a regression-based method with case-based tuning to solve the due date assignment problem

D. Y. Sha [a b] , R. L. Storch [c] & C.-H. Liu [a]

[a] Department of Industrial Engineering and Management , National Chiao Tung University , 1001 Ta Hsueh Road, Hsinchu, Taiwan 30050, Republic of China

[b] Graduate Institute of Business Administration , Asia University , Taichung, Taiwan, Republic of China

[c] Industrial Engineering Program , University of Washington , Box 352650, University of Washington, Seattle, WA 98195-2650, USA

Published online: 14 Oct 2010.

International Journal of Production Research,
Vol. 45, No. 1, 1 January 2007, 65–82

Taylor & Francis
Taylor & Francis Group

Development of a regression-based method with case-based tuning to solve the due date assignment problem

D. Y. SHA*†‡, R. L. STORCH§ and C.-H. LIU†

†Department of Industrial Engineering and Management,
National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu,
Taiwan 30050, Republic of China
‡Graduate Institute of Business Administration, Asia University,
Taichung, Taiwan, Republic of China
§Industrial Engineering Program, University of Washington,
Box 352650, University of Washington, Seattle, WA 98195-2650, USA

(*Revision received December 2005*)

Many regression-based methods to date have been proposed for solving the due date assignment (DDA) problem. The advantages of regression-based DDA methods are that they are easy to both put into practice and comprehend. However, relatively little scheduling research has focused on improving the performances of regression-based DDA methods. The performance of a regression-based DDA method could be improved if its values of regression coefficients could provide a more accurate and precise flowtime estimation for each individual job. The difficulty in doing this stems from the dynamic and stochastic nature of production environment that precludes accurate estimation. Therefore, the aim of this study is to suggest a particular methodology for setting the regression coefficients to improve the performance of regression-based DDA method. In particular, the regression-based DDA method achieved by our suggested methodology is able to adjust the values of coefficients dynamically to best predict the job due date based on the condition of the shop at the instant of job entry. To evaluate the robustness of the methodology, an experimental design was used with four regression coefficient determining procedures, two shop models, and three dispatching rules. The results of this investigation clearly indicate that significant improvements in the performance of regression-based DDA method can occur when the suggested methodology is used.

Keywords: Due date assignment; Regression-based method; Dynamic tuning

1. Introduction

With the current emphasis on the just-in-time (JIT) production philosophy, it is crucial to meet the due dates of jobs. Assigning exact due dates and timely delivery of goods enhances customer satisfaction as well as providing a competitive advantage. Consequently, due date assignment (DDA) is an important task in shop

*Corresponding author. Email: yjsha@mail.nctu.edu.tw

International Journal of Production Research
ISSN 0020–7543 print/ISSN 1366–588X online © 2007 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00207540500507435

floor control. Therefore, the production system must predict the completion time for an order so as to meet the expected due date for the customer. In general, order completion time can be represented by equation (1) (Chang 1994):

$$f_i = r_i + p_i + q_i, \tag{1}$$

where f_i, r_i, p_i, and q_i denote the completion time for order i, the arrival time for order i, the sum of the processing time of all operations for order i, and time allowance for order i, respectively. Terms r_i and p_i are known constants when order i arrives. The time allowance for the order (q_i) is the only variable that needs to be estimated for predicting f_i in equation (1). Hence, the manager must establish an applicable prediction model for q_i to precisely predict the due date of new orders.

In several previous studies, the regression technique was used to develop a due date predictor. The advantages of using a regression-based DDA method are that it is easy both to practice and to comprehend. One of the earliest works is that reported by Conway (1965). Four due date assignment methods were analysed in his study:

(i) constant allowance (CON)
(ii) total work content (TWK)
(iii) common slack (SLK)
(iv) random allowance (RAN)

Among these four methods, TWK is a regression-based DDA method. Eilon and Chowdhury (1976) later compared the following two approaches for setting due date, the first based upon job characteristics (e.g. TWK, SLK, number of operations (NOP)) and the second based upon job characteristics combined with shop status (e.g. jobs in queue (JIQ), jobs in system (JIS)). Ragatz and Mabert (1984) used a variety of job-related and shop-related factors as independent variables to generate an allowance predictor by the multiple regression technique (termed RMR in their study). Vig and Dooley (1991) presented two new methods: congestion and operation flowtime sampling (COFS) and operation flowtime sampling (OFS). These methods estimate job flowtime based on the variation of flowtime from the last three completed orders. Tsai *et al.* (1997), Cheng and Jiang (1998), Roman and Valle (1996), and Chang and Lee (2000) also applied regression analysis models to estimate the job due date.

Regression-based methods can actually be divided into two classes—static and dynamic—depending on whether or not the regression coefficients vary with shop status. For the static method, all input cases adopt the same set of static coefficients throughout the entire due date prediction process. There are two procedures that can be used to construct this kind of method. The first is the single-step (SIG) procedure, in which the values of the regression coefficients are determined through a single regression analysis on either real data or the results of a simulation. The second is the iterative regression (IR) procedure, in which the values of the regression coefficients are determined by the use of regression to an iterative process. The idea behind this procedure is that the due date of a job has an effect upon its flowtime when a due date oriented dispatching rule is used. Therefore, in order to determine more appropriate regression coefficients, the IR procedure addresses the interaction between the job due date and the job flowtime by repeating the simulation-regression sequence until significant improvement is no longer realised. Gee and Smith (1993) showed that the IR procedure for determining the regression coefficients results in better shop performance than a single-step regression procedure in a single-stage job shop. For the dynamic coefficient method, each regression coefficient is adjusted dynamically according to the shop status. Smith (1995) proposed a dynamic version of the regression (DYN) procedure that updates the regression coefficients of the allowance equation every two hundred jobs. Smith's results demonstrated that a dynamic, on-line update of the regression coefficients could provide excellent results. Though the IR and DYN procedures probably provide a considerable advantage, their disadvantage is that the modelling is not easily implemented.

In addition to applying the regression technique to conduct DDA, many researchers have attempted to utilise other techniques, such as analytical methods, simulation methods, capable-to-promise (CTP) methods and machine learning tools. Analytical methods based on queuing theory estimate the mean and standard deviation of order flow time (Enns 1993). Because analytical method assumptions usually conflict with the conditions of the real world, analytical methods are restricted in their application (Chung *et al*. 1997). Advances in computer technology have made simulations into one of the public methods used in DDA research. The simulation-based DDA method performs well as long as the manager has modelled a simulation model in detail. Moses *et al*. (2004) proposed a real-time promising (RTP) method, which is referred to as a CTP method, and showed that the RTP method is superior to six alternative methods for flowtime estimation. In recent years, neural networks (NN) are becoming better known and have been successfully implemented in manufacturing (Udo 1992). Philipoom *et al*. (1994) used neural network models to forecast the job due date in a flow shop manufacturing system. Their research pointed out that NN could outperform conventional regression-based DDA methods. Sha and Hsu (2004a, b) explored NN in the due date assignment problem at a wafer fabrication factory, and their experimental results showed that the NN is highly effective and comparable with conventional regression-based DDA methods. However, NN is regarded as a black-box technology because of the lack of comprehensibility. In our study, comprehensibility is very important whenever the developed DDA method will be used to support a decision made by production managers. If the DDA method is not comprehensible to the user, then the user may not be able to interpret it properly and may not have sufficient trust in the DDA method to use it for due date setting. Therefore, in our study, a modified form of the regression-based method is suggested for DDA. It inherits the comprehensibility of the regression-based method and high prediction accuracy of NN in order to support a production manager in predicting the job due dates both efficiently and precisely.

Our purpose here is to begin the recommendation of a methodology for estimating the values of regression coefficients of the regression-based DDA method and attempt to see if the methodology is able to improve the scheduling performance of the regression-based method. This methodology captures most of the benefits of the regression procedure for setting due dates, is more responsive to changing shop status when a job arrives and is easily implemented. Our suggested methodology is based on a machine learning tool, the model trees algorithm. The model tree is a new technology for predicting a numeric value that was developed by Quinlan (1992) and has been successfully applied to solve the classification problem.

To the best of our knowledge, the use of model trees for designing a regression model for setting the due date is a research area still unexplored. The regression-based DDA method achieved by our suggested methodology—namely, regression-based method with case-based tuning (R-CBT)—has the capability of adjusting the values of the regression coefficients dynamically to accurately set the job due date according to the shop conditions when the job arrives. For comparison purposes, the single-step, iterative regression and dynamic version procedures were used to construct three multiple-variables regression-based DDA methods as the benchmarking methods. And we also conducted an experiment to compare the performance and efficiency to two DDA methods (Neural network-based predictor and R-CBT).

2. Research problem

Our research question may be stated as follows: does the methodology for determining regression coefficients suggested in this study improve scheduling performance compared to using the single-step (SIG), iterative regression (IR) or dynamic version (DYN) procedures? In order to test the robustness of the suggested regression-based DDA method, several dispatching rules and shop models were considered. This section is divided into four subsections: dispatching rule, shop model, regression coefficients determining procedure and performance measure.

2.1 *Dispatching rule*

Three typical dispatching rules are listed in order of increasing sophistication in the use of information to determine the next job to be processed on an available machine: first come first served (FCFS), earliest due date (EDD), and shortest processing time (SPT). We chose those rules because they do not need parameter estimation, are most frequently used in previous studies, and have different characteristics from one another. Among these rules, EDD is a due date oriented rule, SPT is a process-time oriented rule, and FCFS is a random rule that was used as the base-line rule in the experiments.

2.2 *Shop model*

Our attention also turned to an investigation of the effect of shop structure. Naturally, one could not hope to generalise results across all shops, because of the wide range of operating conditions in real-world environments (Philipoom 1994). Rather, we modestly investigated two shop models, one wafer fabrication factory and one a job shop. Both shop models were built on a personal computer with a Pentium III 1.3 GHz processor using eM-Plant 4.6, a simulation package developed by Tecnomatix Technologies Ltd.

2.2.1 Wafer fabrication factory. The configuration of the wafer fab considered in this study is the same as that of the Hewlett-Packard Technology Research Center Silicon fab (hereafter referred to as the TRC fab) studied in Wein (1988) and Kim (1998). This facility is a relatively large development laboratory in

Palo Alto, California. The TRC fab consists of 24 single-server or multi-server stations, with all multi-server stations consisting of identical pieces of equipment. In this fab, the bottleneck workstation is the photographic expose station (station 14), which is utilised much more than other stations. Stations 1 to 4 are batch processing stations in which a set of queued jobs can be processed together. These stations operate under the minimum batch size (MBS) strategy, which is widely practised. Under this strategy, a batch is started as soon as there are at least a specified MBS number of lots available, and an available capable machine. In this study, processing starts at these four stations when the number of waiting lots is greater than or equal to two.

This test model has five products. The five products flow through the photographic expose station 14, 9, 13, 12 and 12 times respectively. The processing time for a lot at station j in our simulation study is randomly generated from uniform (0.9 p_j, 1.1 p_j), where p_j is the mean processing time (MPT) of a lot at station j. Set-up time is included in the processing time, and the transfer time between stations is ignored. The inter-arrival times of orders are generated from an exponential distribution and are determined using a preliminary experiment in which bottleneck station utilisation is nearly 90%. The simulation model also takes into account machine failures, including unscheduled breakdowns and scheduled maintenance. Time between failures and time to repair for each workstation are randomly generated from exponential distributions with the mean time between failures (MTBF) and the mean time to repair (MTTR). The parameters (MPT, MTBF, and MTTR) used to generate the processing times, time between failures, and time to repair for each station were given in Kim (1998).

2.2.2 Job shop. This research used a 10×10 benchmark problem from Lawrence (1984). This test model has 10 jobs, each with 10 operations and 10 machines. Table 1 provides the data for the problem using the following structure: machine, processing time. In this study, the probability of each product being chosen to be released into the shop is equal. The inter-arrival times of jobs are generated from a negative exponential distribution, which has a mean value chosen to create a certain expected shop utilisation rate of 90%.

Table 1. Job shop model (Lawrence 1984).

	Operation									
Job	1	2	3	4	5	6	7	8	9	10
1	5, 18	8, 21	10, 41	3, 45	4, 38	9, 50	6, 84	7, 29	2, 23	1, 82
2	9, 57	6, 16	2, 52	8, 74	3, 38	4, 54	7, 62	10, 37	5, 54	1, 52
3	3, 30	5, 79	4, 68	2, 61	9, 11	7, 89	8, 89	1, 81	10, 81	6, 57
4	1, 91	9, 8	4, 33	8, 55	6, 20	3, 20	5, 32	7, 84	2, 66	10, 24
5	10, 40	1, 7	5, 19	9, 7	7, 83	3, 64	6, 56	4, 54	8, 8	2, 39
6	4, 91	3, 64	6, 40	1, 63	8, 98	5, 74	9, 61	2, 6	7, 42	10, 15
7	2, 80	8, 39	9, 24	4, 75	5, 75	6, 6	7, 44	1, 26	3, 87	10, 22
8	2, 15	8, 43	3, 20	1, 12	9, 26	7, 61	4, 79	10, 22	6, 8	5, 80
9	3, 62	4, 96	5, 22	10, 5	1, 63	7, 33	8, 10	9, 18	2, 36	6, 40
10	2, 96	1, 89	6, 64	4, 95	10, 23	8, 18	9, 15	3, 64	7, 38	5, 8

(a, b): (machine, processing time).

2.3 *Regression coefficients determining procedure*

Three regression coefficient determining procedures were used in this study as benchmarking procedures: single-step, iterative regression and dynamic version procedures. The most commonly used procedure is the single-step (SIG) procedure, in which the values of the regression coefficients of the regression equation are determined through only one regression analysis on the results of a simulation or real data. This procedure's advantage is that it is easy to implement because there is only one regression analysis needed.

Since the due date of a job has an effect on its flowtime when a due date oriented dispatching rule is used, Gee and Smith (1993) proposed a iterative regression (IR) procedure to determine more appropriate regression coefficients, in which the values of the regression coefficients are determined by applying regression to an iterative process. The IR procedure addresses this interaction by repeating the simulation-regression sequence until a significant improvement of performance of the regression-based DDA method is no longer realised. The steps of the IR procedure are as follows: first, a single regression analysis is applied to the initial simulation data to generate the first-cycle coefficient values for the regression equation; the second step is an iterative step repeating the simulation-regression sequence. In each simulation, the job stream is the same as the one that used to generate the initial simulation data. That means that it uses the same job stream to both test the performance of the regression equation that was generated from the previous cycle and produce the new training data to generate the next-cycle regression equation. The second step repeats until there is no significant improvement in due date related performance between the current regression equation and the prior-cycle regression equation.

Smith (1995) proposed a dynamic version of the regression coefficients generating (DYN) procedure that updated the values of the regression coefficients of the regression equation every two hundred jobs. In the DYN procedure, the initial regression equation is generated from the initial training data, but the values of regression coefficients are updated every specified number of jobs when it is predicting the unseen job due dates. With the actual data from the most recent set of jobs providing the values of the regression coefficients of regression equation, the new equation is used to predict job due dates for the next set of unseen jobs.

2.4 *Performance measure*

The quality of the job due date predictor can be determined in terms of 'accuracy' and 'precision'. Vig and Dooley (1993) defined the accuracy of an estimate as how close the individual estimates were to their true values and the precision was defined as the variability of prediction errors. In this study, we used mean absolute lateness (MAL) to measure the accuracy and mean squared lateness (MSL) to measure the precision. The formulae used for the performance measure are as follows.

$$\mathrm{MAL} = \sum_{i=1}^{n} [\max(0, d_i - f_i) + \max(0, f_i - d_i)]/n, \tag{2}$$

$$\mathrm{MSL} = \sum_{i=1}^{n} [\max(0, d_i - f_i) + \max(0, f_i - d_i)]^2/n, \tag{3}$$

Here, f_i, d_i, and n denote the completion time, promised due date of job i, and sample sizes, respectively.

3. Methodology

This section discusses in detail the suggested methodology for determining the values of the regression coefficients for the allowance equation. This methodology is based on the model trees algorithm, so we will introduce it in detail. Then the framework of R-CBT, as achieved by model trees, is described. This section is divided into two subsections—model trees and framework of R-CBT.

3.1 *Model trees*

A new technique for predicting the value for a test instance, the model trees (M5), was developed by Quinlan (1992). There are three major stages of model tree construction. The first generates the tree-based model in a top-down recursive divide-and-conquer manner. At the beginning, all the training cases are at the root node. Cases are partitioned recursively based on choosing the attribute-value pair that does the best job in discriminating cases, so as to make the activity levels in the subset more homogeneous. For each non-terminal node in the tree-based model, M5 divides all cases at that node into two subsets of cases corresponding to test outcomes of each potential attribute-value pair at that node and then calculates the expected reduction in error. The expected reduction in error can be written as equation (4) and is given in Quinlan (1992). After examining all potential tests, the attribute-value pair that maximises the expected reduction in error is chosen to split the cases at that node.

$$\Delta\text{error} = sd(T) - \sum_i \frac{|T_i|}{|T|} \times sd(T_i), \tag{4}$$

where T is the set of cases that reach this node, T_1 and T_2 are the subset of cases that result from splitting the node according to the selected attribute-value pair (e.g. $\text{case}_j \in T_1$ if attribute $(x) \leq C$; $\text{case}_j \in T_2$ if attribute $(x) > C$), and the sd function calculates the standard deviation of the target values of a set of cases. The process of splitting the node terminates when very few cases are contained or the target values of cases that reach a node vary only slightly.

Second, after the tree-based model is built, a multivariate linear model is implemented for the cases at each node of the tree-based model using the standard regression technique. However, instead of using all attributes, this linear model is restricted to the attributes referenced by the tests or by linear models somewhere in the subtree at this node (Quinlan 1992). In order to minimise the linear models' estimated error in unseen cases, these models are simplified by eliminating parameters. Because the elimination of parameters generally causes the average residual coming from a set of training cases to increase, M5 multiplies the value by $(n+\nu)/(n-\nu)$, where n is the number of training cases at that node and ν is the number of parameters in the linear model. The third stage prunes the tree, starting near the bottom. For each non-leaf node of the tree, M5 selects as the final model either the simplified linear model above or the subtree, whichever has the lower estimated error (Quinlan 1992).

When a tree-based model is built completely to predict the value of an unseen case, the tree is followed down a leaf, according to the attribute values of the case, to make routing decisions at each node. The leaf will involve a linear model based on some attributes, and the value of the case is predicted by that linear model to yield a raw predicted value. Quinlan (1992) used a smoothing process to compensate for the sharp discontinuities. The smoothing process filters the predicted value at each node along the path from the leaf to the root. An appropriate smoothing calculation is given in Wang and Witten (1997) and is shown in equation (5).

$$p' = \frac{np + kq}{n + k}, \tag{5}$$

where p' is the prediction passed up to the next higher node, p is the prediction passed to this node from below, q is the value predicted by the model at this node, n is the number of cases that reach the node below, and k is a smoothing constant (default value 15).

A slightly different representation for the smoothing process, shown in Frank *et al.* (1998), has the same effect as the above procedures, in which a new linear model is created at each leaf that combines the linear models along the path back to the root, in order to let the leaf models automatically create smoothed predictions without any need for further adjustment at the time predictions are made. Frank *et al.* (1998) showed an example of this implementation, in which they supposed that the model at a leaf involved two attributes x and y, with linear coefficients a and b; and the model of a leaf at the parent node involved two attributes y and z, with linear coefficients c and d:

$$p = ax + by \quad q = cy + dz.$$

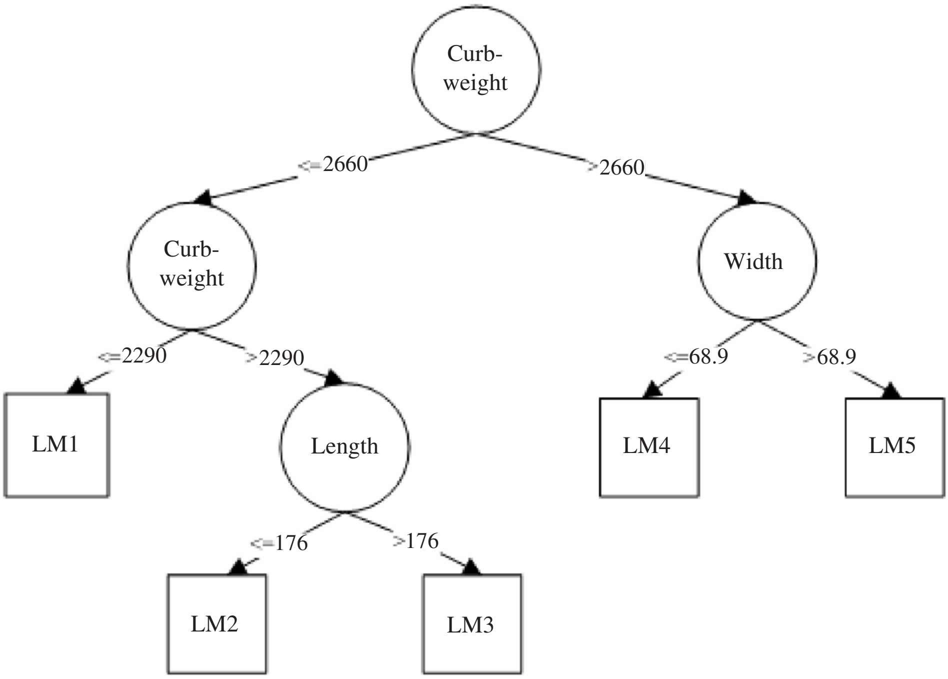
They then combined these models into a single model using equation (6):

$$p' = \frac{na}{n + k}x + \frac{nb + kc}{n + k}y + \frac{kd}{n + k}z. \tag{6}$$

For each leaf, a single, smoothed linear model can be achieved by continuing in this way up from the leaf to the root, and this linear model directly yields a predicted value (Frank *et al.* 1998).

In summary, model trees are binary decision trees with linear regression functions at the leaf node. Thus, they can represent any piecewise linear approximation to an unknown function (Frank *et al.* 1998). Model trees have advantages in prediction accuracy over regression trees (Quinlan 1992) and over linear regression (Quinlan 1993a, b, Quinlan 1992, Wang and Witten 1997). A rational reconstruction of M5, called M5′, was described by Wang and Witten (1997) along with further implementation details, such as dealing effectively with enumerated attributes and with missing values. M5′ seems to outperform M5. Therefore, M5′ was used in this study.

Figure 1 shows the structure of the model trees found by M5′ for the data *autoprice*. This data contains 15 continuous variables covering quantities such as automobile length, weight, and engine size in order to predict the 1985 list price



Model	LM1	LM2	LM3	LM4	LM5
Constant term	–15200	–7700	–10700	–6570	–16700
Symboling					
Normalised losses	7.19	8.42	34.6		
Wheelbase	37	56.4	56.4		
Length	–10.7	–10.7	–10.7	–20.4	–20.4
Width	121	121	121	230	230
Height					
Kerb weight	4.69	1.76	1.76	3.46	9.56
Engine size					
Bore					
Stroke					
Compression ratio				124	244
Horsepower	23.4	9.62	9.62	32.4	6.22
Peak-rpm					
City mpg	–62			–211	–415
Highway mpg	71.2				

Figure 1. Model tree and linear models for data set *autoprice*.

of common automobiles. Each item in the table is the regression coefficient of the smoothed linear model.

3.2 *Framework of R-CBT*

The regression-based DDA method studied here is shown in equation (7) and is a multiple-variables regression equation, where $\hat{f}_i, r_i, P_i$ and $F_i\ (\cdot)$ denote the estimated completion time, the arrival time, the total processing time and allowance prediction equation of job i, respectively. The basic concept of this equation is that the allowance is a complex function of many shop and job characteristics. Therefore, for this study we attempted to collect as many important independent variables $(x_{i1}, x_{i2}, \ldots, x_{in})$ as possible to construct an allowance predictor to predict job allowance.

$$\hat{f}_i = r_i + p_i + F_i(x_{i1}, x_{i2}, \ldots, x_{in}), \tag{7}$$

For each combination of dispatching rule and shop model, we gathered sufficient data in order to determine the prediction variables and the values of regression coefficients of the above equation. In the job shop, for each selected job, five general job characteristics were obtained as well as 42 shop statuses, resulting in 47 characteristics per selected job (see table 2). In the wafer factory, five job characteristics and 21 shop statuses, resulting in 26 characteristics, were collected per selected job (see table 3). The differences between tables 2 and 3 are the result of three factors. First, the total number of products in the wafer factory is different from those in the job shop. Second, in order to reduce the cost of data acquisition, station information regarding the loads and queue line length was only gathered from the 1st, …, 3rd bottleneck stations in the wafer factory. Third, the information regarding machine shutdowns in the 1st, …, 3rd bottleneck stations was specifically collected in the wafer factory. After data collection, each datum included both job and shop characteristics, and the actual allowance throughout the shop for each collected job.

Table 2. The 47 variables for each selected job in job shop.

Variables	Information
Job characteristics	
B1, …, B3	Processing times for operating in 1st, …, 3rd bottleneck machine for job i
TW	Sum of processing times for job i
TYPE	The type of job
Shop characteristics	
M1QL, …, M10QL	Number of jobs presently in queue of machine 1, …, machine 10
M1WL, …, M10WL	Total remaining workload of the machine 1, …, machine 10 for all the jobs in the shop
NJ1, …, NJ10	Total work in process of job 1, …, job 10 in the shop
J1R, …, J10R	Average flow time (three lots) of job 1, …, job 10, which had most recently completed jobs
SRT	Sum of the remaining processing time for all jobs in the shop
WIP	Work in process in the shop

Table 3. The 26 variables for each selected job in wafer factory.

Variables	Information
Job characteristics	
B1, ..., B3	Processing times for operating in 1st, ..., 3rd bottleneck station for job i
TW	Sum of processing times for job i
TYPE	The type of job
Shop characteristics	
M1QL, ..., M3QL	Number of jobs presently in queue of 1st, ..., 3rd bottleneck station
M1WL, ..., M3WL	Total remaining workload of 1st, ..., 3rd bottleneck station for all the jobs in the shop
M1D, ..., M3D	The number of shutdown machine in 1st, ..., 3rd bottleneck station
NJ1, ..., NJ5	Total work in process of job 1, ..., job 5 in the shop
J1R, ..., J5R	Average flow time (three lots) of job 1, ..., job 5, which had most recently completed jobs
SRT	Sum of the remaining processing time for all jobs in the shop
WIP	Work in process in the shop

As mentioned above, many variables were collected simultaneously, some of which influenced the allowance, while others did not. Identifying the proper set for a given shop and including these variables explicitly in the allowance estimates should improve the accuracy of prediction (Philipoon *et al.* 1994). Therefore, instead of using the raw data to develop a due date predictor directly, this study acquired influential variables by means of screening by using a stepwise regression procedure on the raw data. Once the influential variables were identified, each datum in the raw data was represented by these variables and by their actual allowance. The training data was then fed into the model trees to develop the R-CBT for improving prediction accuracy. When a R-CBT was built completely to predict the due date of an unseen case, the case was represented using the row vector that had the values of the influential variables that had been identified by the above procedure, and the tree of R-CBT was followed down to a leaf according to the values of variables of the case to make routing decisions at each node. This leaf involved a specific linear model based on some of the variables, and the unseen case was evaluated by the linear model so as to yield a predicted job due date. The framework of the R-CBT is shown in figure 2.

4. Experimental procedure

To evaluate the robustness of different DDA methods, two shop models—a wafer factory and a job shop—and three dispatching rules—FCFS, SPT, and EDD—were studied. There were two stages to the experimental procedure. The first was the modelling stage, and the other was the testing stage.

In the first stage, for each combination of shop and dispatching rule, we collected data on the allowance of jobs through the shop, taking note of shop conditions at the time of job entry as well as job characteristics (mentioned in section 3.2). Because the data was then used to estimate the values of coefficients in the

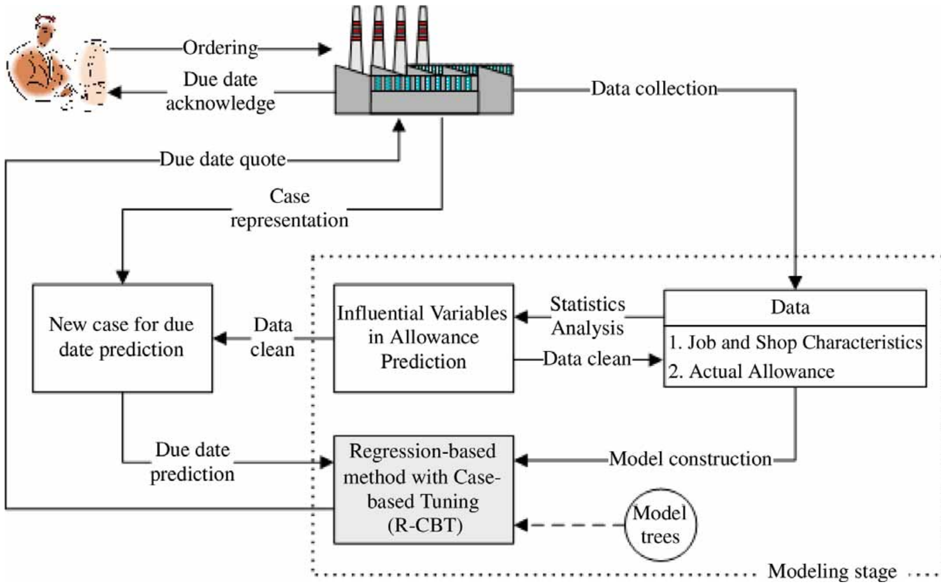


Figure 2. The framework of R-CBT.

regression models, it was necessary to guarantee statistical independence among the cases before the test was performed. To ensure this, once the shop reached the steady state, only 1 in every 30 outputs from the shop was randomly selected to be included in the sample of 1000 jobs as the raw training data. After performing the feature selection step, each datum in the raw training data was represented by these influential variables and by their actual allowance. R-CBT and three multiple-variables regression-based DDA methods achieved by three regression coefficient determining procedures (SIG, IR, and DYN procedures) were based on the same training data. These three benchmark methods are called SIGM, IRM and DYNM. By comparing the performance of R-CBT with the performances of SIGM, IRM and DYNM, we ensured a level playing field in which the four types of regression-based DDA methods used the same information to evaluate the effectiveness of different regression coefficient determining procedures. Specifically, the initial allowance equation of DYNM was generated from the cleaned training data, but its value of regression coefficients was updated every 500 jobs in our study when it predicted the unseen cases' due date. With the actual allowance of the most recent 500 jobs providing the value of regression coefficients of the allowance equation of DYNM, the new equation predicted job allowances for setting the job due date for the next 500 jobs. In order to construct IRM, a regression analysis was first applied to the cleaned training data to produce the first-cycle coefficient values for the allowance equation. The iterative regression procedure then repeated the simulation-regression sequence, in which the simulation used the same job stream that was used in the modelling stage to test the performance of allowance equation generated from the prior-cycle and to produce the new training data for generating the next-cycle allowance equation, until significant improvement was no longer realised. Since there was no interaction between the job due date and the job flowtime when a non due date oriented dispatching rule was used (i.e. SPT and FCFS), the IRM was specifically adopted when the EDD rule was used. Two simplified regression-based DDA methods, TWK and JIQ, were also selected as benchmark methods. The values of regression coefficients of both methods were determined from a regression analysis (SIG procedure) on the same training data.

In the second stage of the experimental procedure—namely, the testing stage—ten replications were conducted based on the allowance equations determined in the modelling stage, and common job streams were used within each replication. The job stream used in the modelling stage was not among the 10 job streams used in the testing stage. The use of common job streams resulted in similar experimental conditions for the comparison of different DDA methods. For each replication, once the shop reached the steady state, the information was then collected on the two performance measures (MAL, MSL) for the next 1000 jobs.

5. Results

This section is divided into two subsections—the fundamental experiment and the R-CBT vs neural networks. In the first subsection, we describe how, for each existing regression coefficient determining procedure tested, two due date-related performance measures were collected and compared. In the latter subsection, having demonstrated that model trees generally provide the methodology of choice for estimating the values of regression coefficients, our attention turns to an investigation of the performance comparison between a neural networks-based predictor and R-CBT.

5.1 *Fundamental experiment*

Figure 3 shows the compositions of R-CBT, SIGM, IRM, DYNM, JIQ, and TWK for the combination of the wafer factory and the EDD rule. Four influential

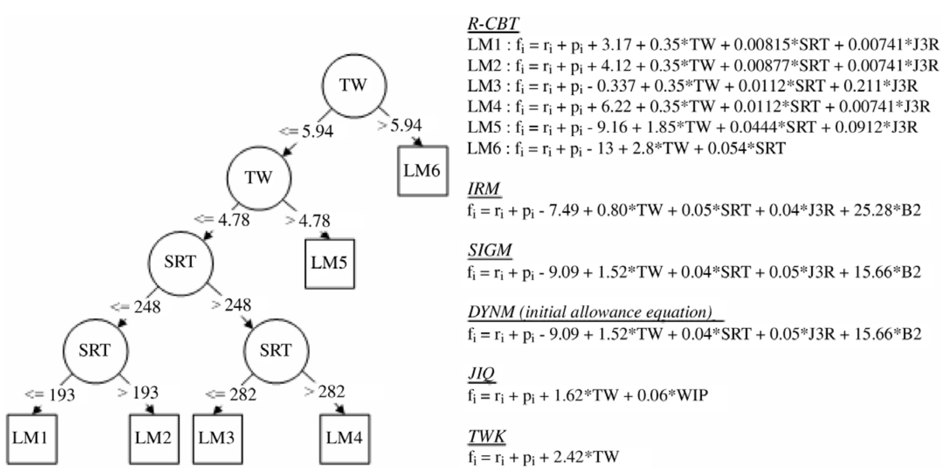


Figure 3. The composition of all DDA methods.

variables, TW, SRT, J3R, and B2, are indicated in this case. As illustrated in figure 3, six linear models were used in R-CBT, and each model had specific values of regression coefficients. When a new job arrived, a specific linear model was chosen from the available options to estimate the job due date based on the values of the influential variables at the instant of job entry.

The results of the factorial experiment are summarised in tables 4 and 5. Each item in the tables is an average of the experiment's 10 replications. The best result for the due date assignment problem is highlighted in boldface and italic. Every comparison, except for the FCFS rule in the job shop, finds that R-CBT outperforms all of other multiple-variables regression-based DDA methods (i.e. SIGM, DYNM and IRM), which means that the performance of the regression-based DDA method is improved by using model trees to estimate the values of coefficients. The one exception is the FCFS rule in the job shop, in which the same regression equation was used in R-CBT and SIGM, so the performance of R-CBT is the same as the performance of SIGM. These tables also illustrate that R-CBT performs better than JIQ and TWK with respect to MAL and MSL in all combinations of dispatching rule and shop, and the relative performances of IRM, DYNM, and SIGM depend on the dispatching rule used. In addition, for both shops, the overall best performance is obtained with the R-CBT in conjunction with the EDD rule, and as such it provides the greatest improvement with respect to due date related performance.

The main objective of this study is to study the relative effects of various regression coefficient determining procedures and DDA methods in different scheduling combinations. Because common job streams were used to generate the 10 observations in each cell, the sample observations were not independent.

Table 4. Performance of selected DDA methods under different dispatching rules in wafer factory.

		Performance measure	
Dispatching rule	Due date setting	MAL	MSL
EDD	R-CBT	***1.75***	***4.86***
	IRM	1.98	6.27
	DYNM	2.04	6.63
	SIGM	1.99	6.80
	JIQ	2.46	9.16
	TWK	3.02	12.98
SPT	R-CBT	***3.42***	***26.74***
	DYNM	4.19	40.07
	SIGM	4.45	42.75
	JIQ	7.48	94.10
	TWK	7.44	93.64
FCFS	R-CBT	***2.33***	***8.70***
	DYNM	2.42	9.41
	SIGM	2.44	9.35
	JIQ	2.82	12.18
	TWK	3.89	21.34

Time unit: day.

Table 5. Performance of selected DDA methods under different dispatching rules in job shop.

Dispatching rule	Due date setting	Performance measures MAL	Performance measures MSL
EDD	R-CBT	***121.43***	***23567.98***
	IRM	127.63	26137.71
	DYNM	131.51	27980.43
	SIGM	129.88	27026.62
	JIQ	142.58	31827.14
	TWK	269.73	113713.62
SPT	R-CBT	***296.49***	***463324.15***
	DYNM	346.45	569840.70
	SIGM	350.70	544276.16
	JIQ	357.37	645487.31
	TWK	350.78	685227.84
FCFS	R-CBT	***149.92***	***37393.65***
	DYNM	155.53	41003.05
	SIGM	***149.92***	***37393.65***
	JIQ	171.65	48236.57
	TWK	371.43	234019.46

Time unit: second.

Table 6. Comparison of DDA methods.

Performance measure	Shop	Dispatching rule	Due date assignment method					
MAL	Wafer factory	EDD	R-CBT	IRM	SIGM	DYNM	JIQ	TWK
		SPT	R-CBT	DYNM	SIGM	TWK	JIQ	
		FCFS	R-CBT	DYNM	SIGM	JIQ	TWK	
	Job shop	EDD	R-CBT	IRM	SIGM	DYNM	JIQ	TWK
		SPT	R-CBT	DYNM	SIGM	TWK	JIQ	
		FCFS	R-CBT	SIGM	DYNM	JIQ	TWK	
MSL	Wafer factory	EDD	R-CBT	IRM	SIGM	DYNM	JIQ	TWK
		SPT	R-CBT	DYNM	SIGM	TWK	JIQ	
		FCFS	R-CBT	SIGM	DYNM	JIQ	TWK	
	Job shop	EDD	R-CBT	IRM	SIGM	DYNM	JIQ	TWK
		SPT	R-CBT	SIGM	DYNM	JIQ	TWK	
		FCFS	R-CBT	SIGM	DYNM	JIQ	TWK	

As a result, it was essential to use a paired statistics test (the paired *t*-test) to detect significant statistical differences in the performance of every pair of DDA methods. In order to achieve an experiment-wise confidence level of 95%, we used the Bonferroni approach to control each confidence level for each comparison. Table 6 represents the results of the paired *t*-tests. These DDA methods in table 6 are listed in descending order of performance. They are grouped into homogeneous subsets that are indicated by an underline if the difference between the means of performance measure of the two methods in the subset is not significantly beyond the prescribed α level. Under all combinations of dispatching rule and shop model, except for the FCFS rule in the job shop, R-CBT is significantly better than all of other methods according to the rule ranking listed in table 6. When the FCFS rule is used in the job shop, the differences between R-CBT, DYNM, and SIGM are not significant in MAL and the differences between R-CBT, DYNM, SIGM, and JIQ are not significant in MSL. In summary, the findings show that marked improvements in the due date related performance occurs when the model trees algorithm is used to develop a regression-based DDA method with case-based tuning.

5.2 *R-CBT vs. neural networks*

As a final investigation, the performances of R-CBT were compared with the due dates determined by neural networks. The back-propagation neural network (BPN), which is widely used and produce good prediction results and the pattern recognition BPN model, was used in our study for order due date prediction. The only combination used in this experiment was at the wafer factory with the SPT rule. This was due to the fact that a wafer factory with the SPT rule creates a more dynamic environment, which made it difficult to estimate the job flowtime. The same procedure described above was followed to generate training data and to test the performances of both methods. The average values for performance measures are reported in table 7. The fourth column in table 7 indicates the training time for constructing the prediction model. The differences of MAL and MSL between the R-CBT's and neural networks' results are in general very small, but the TIME measure shows significant differences in favour of R-CBT. The results of the paired t-tests illustrate that there is no significant difference between R-CBT and NN with respect to MAL or MSL (p-value = 0.14 and p-value = 0.37). However, these results are encouraging. The efficiency of R-CBT is clearly demonstrated, because R-CBT requires less time for training without precluding accurate predictions. The fact that R-CBT performs as well as the neural networks is considered a success since it is simpler, requires less training time, and is easier to comprehend.

6. Summary and conclusions

In this study we have suggested a methodology for estimating the values of coefficients of the regression-based DDA method. The proposed methodology is based on using model trees to develop a tree-based piecewise linear regression model as a due date predictor. This novel regression-based DDA method is called

Table 7. Comparison of R-CBT vs. Neural networks.

DDA method	MAL	MSL	TIME
R-CBT	3.42	26.74	0.23
Neural networks	3.35	27.24	3.84

the regression-based method with case-based tuning (R-CBT). This method is able to dynamically choose the most appropriate allowance equation with specific regression coefficients and prediction variables for due date setting, based on the condition of the shop at the instant of job entry.

First-stage experiments in our study tested the performance of different regression coefficients' determining procedures. We conducted a factorial design with three factors (regression coefficient determining procedure, dispatching rule, and the shop model). Thereafter, R-CBT was compared with the multiple-variables regression-based methods achieved by three other popular procedures (single-step, iterative regression, and dynamic version procedures) and two simplified commonly-used DDA methods (JIQ and TWK) using the single-step procedure. Two performance measures (MAL and MSL) were used to compare the DDA methods. Under all combinations of dispatching rule and the shop model, except for the FCFS rule in the job shop, R-CBT exhibited excellent due date performance by providing more accurate and precise estimations of job flowtime than three multiple-variables regression-based DDA methods (i.e. SIGM, IRM and DYNM), JIQ and TWK. The findings of our experiments clearly show that the greatest improvement for the regression-based DDA method can be obtained by using our suggested methodology to determine the values of regression coefficients. Second-stage experimental results demonstrated that there is no significant difference in the due date related performance of R-CBT and the neural networks, but R-CBT can be quickly modelled and is easier to implement and to comprehend than neural networks.

As a further research topic, even though dynamic information such as machine shutdown was considered when developing R-CBT, regression coefficients in R-CBT could also be updated as time passes. However, this would require establishing a system to monitor the residual results from R-CBT and to advise the production manager when to re-train R-CBT in order to maintain the prediction accuracy of R-CBT. In addition, an obvious area for future research is to illustrate and modify our R-CBT to solve the DDA problem with asymmetric earliness-tardiness cost.

References

Chang, F.C.R., A study of factors affecting due-date predictability in a simulated dynamic job shop. *J. Manuf. Syst.*, 1994, **13**, 393–400.

Chang, Y.J. and Lee, C.E., A bottleneck based due date assignment methodology. *Int. J. Manuf. Tech. Manage.*, 2000, **1**, 318–327.

Cheng, T.C.E. and Jiang, J., Job shop scheduling for missed due-date performance. *Comp. & Indust. Eng.*, 1998, **34**, 297–307.

Chung, S.H., Yang, M.H. and Cheng, C.M., The design of due date assignment model and the determination of flow time control parameters for the wafer fabrication factories. *IEEE Trans. on Comp., Pack., Manuf. Tech.—Part C*, 1997, **20**, 278–287.

Conway, R.W., Priority dispatching and job lateness in a job shop. *J. Indust. Eng.*, 1965, **16**, 228–237.

Eilon, S. and Chowdhury, I.G., Due date in job shop scheduling. *Int. J. Prod. Res.*, 1976, **14**, 223–238.

Enns, S.T., Job shop flow time prediction and tardiness control using queue analysis. *Int. J. Prod. Res.*, 1993, **31**, 2045–2057.

Frank, E., Wang, Y., Inglis, S., Holmes, G. and Witten, I.H., Using model trees for classification. *Mach. Learn.*, 1998, **32**, 63–76.

Gee, E.S. and Smith, C.H., Selecting allowance policies for improved job shop performance. *Int. J. Prod. Res.*, 1993, **31**, 1839–1852.

Hsu, S.Y. and Sha, D.Y., Due date assignment using artificial neural networks under different shop floor control strategies. *Int. J. Prod. Res.*, 2004, **42**, 1727–1745.

Kim, Y.D., Kim, J.U., Lim, S.K. and Jun, H.B., Due-date based scheduling and control policies in a multiproduct semiconductor wafer fabrication facility. *IEEE Trans. Semicon. Manuf.*, 1998, **11**, 155–164.

Lawrence, S., Resource constrained project scheduling: an experimental investigation of heuristics scheduling techniques. Graduate School of Industrial Administration, Pittsburgh, Carnegie Mellon University, 1984.

Moses, S., Grant, H., Gruenwald, L. and Pulat, S., Real-time due-date promising by build-to-order environments. *Int. J. Prod. Res.*, 2004, **42**, 4353–4375.

Philipoom, P.R., Rees, L.P. and Wiegmann, L., Using artificial neural networks to determine internally-set due date assignment for shop scheduling. *Decision Sci.*, 1994, **25**, 825–847.

Philipoom, P.R., Wiegmann, L. and Rees, L.P., Cost-based due-date assignment with the use of classical and neural-network approaches. *Naval Res. Logist.*, 1997, **44**, 21–46.

Quinlan, J.R., Learning with continuous classes, in *Proceedings of 5th Australian Joint Conference on Artificial Intelligence*, World Scientific, Singapore, 1992, pp. 343–348.

Quinlan, J.R., A case study in machine learning, in *Proceedings of 16th Australian Computer Science Conference*, Brisbane, 1993a, pp. 731–737.

Quinlan, J.R., Combining instance based and model based learning, in *Proceedings of the 10th International Conference on Machine Learning*, Morgan Kaufmann, San Mateo, CA, 1993b, pp. 236–243.

Ragatz, G.L. and Mabert, V.A., A simulation analysis of due date assignment rules. *J. Op. Manage.*, 1984, **5**, 27–39.

Roman, D.B. and Valle, A.G., Dynamic assignation of due dates in an assembly shop based in simulation. *Int. J. Prod. Res.*, 1996, **34**, 1539–1554.

Sha, D.Y. and Hsu, S.Y., Due-date assignment in wafer fabrication using artificial neural networks. *Int. J. Adv. Manuf. Tech.*, 2004, **23**, 768–775.

Smith, C.H., Minor, E.D. and Wen, H.J., Regression based due date assignment rules for improved assembly shop performance. *Int. J. Prod. Res.*, 1995, **33**, 2375–2385.

Tsai, C.-H., Chang, C.-T. and Li, R.-K., Integrating order release control with due-date assignment rules. *Int. J. Prod. Res.*, 1996, **35**, 3379–3392.

Udo, G.J., Neural networks applications in manufacturing process. *Comp. Indust. Eng.*, 1992, **23**, 97–100.

Vig, M.M. and Dooley, K.J., Mixing static and dynamic estimates for due date assignment. *J. Op. Manage.*, 1993, **11**, 67–79.

Vig, M.M. and Dooley, K.J., Dynamic rules for due date assignment. *Int. J. Prod. Res.*, 1991, **29**, 1361–1377.

Wang, Y. and Witten, I.H., Induction of model trees for predicting continuous classes, in *Proceedings of the Poster Papers of the European Conference on Machine Learning*, University of Economics, Faculty of Informatics and Statistics, Prague, 1997, pp. 128–137.

Wein, L.M., Scheduling semiconductor wafer fabrication. *IEEE Trans. Semicon. Manuf.*, 1988, **1**, 115–130.