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

Development and evaluation of a tree-indexing approach to improve case-based reasoning: illustrated using the due date assignment problem

D. Y. Sha [a] [b] & C.-H. Liu [a]

[a] Department of Industrial Engineering and Management , National Chiao Tung University , 1001 Ta Hsueh Road, Hsinchu, Taiwan , R.O.C. 30050

[b] Graduate Institute of Business Administration , Asia University , Taichung, Taiwan , R.O.C. 41354

Published online: 11 Feb 2011.

To cite this article: D. Y. Sha & C.-H. Liu (2006) Development and evaluation of a tree-indexing approach to improve case-based reasoning: illustrated using the due date assignment problem, International Journal of Production Research, 44:15, 3033-3049, DOI: 10.1080/00207540500168238

To link to this article: http://dx.doi.org/10.1080/00207540500168238

International Journal of Production Research,
Vol. 44, No. 15, 1 August 2006, 3033–3049

Taylor & Francis
Taylor & Francis Group

Development and evaluation of a tree-indexing approach to improve case-based reasoning: illustrated using the due date assignment problem

D. Y. SHA*†‡ and C.-H. LIU†

†Department of Industrial Engineering and Management, National Chiao Tung University,
1001 Ta Hsueh Road, Hsinchu, Taiwan, R.O.C. 30050
‡Graduate Institute of Business Administration, Asia University, Taichung, Taiwan, R.O.C. 41354

(*Received October 2004*)

In this study a novel case indexing approach is proposed for case-based reasoning (CBR). This new approach, called the tree-indexing approach, is a modified form of the inductive learning-indexing (IL-indexing) approach and is especially applied to assist CBR in numeric prediction. The tree-indexing approach organizes the cases in the memory by inducting a tree-shaped structure, in order to improve the efficiency and effectiveness of case retrieval. The experiments, using three real world problems from the UCI repository, show that the CBR with the tree-indexing approach (T-CBR) is superior to the conventional CBR. This study also applies T-CBR for solving the due date assignment problem in a dynamic job shop environment in order to investigate whether T-CBR's expected benefits can be observed in practice. The results of the experiments show that our proposed T-CBR can indeed more accurately predict the job due date than the other methods presently in use.

Keywords: Case-based reasoning; Numeric prediction; Due date assignment

1. Introduction

Case-based reasoning (CBR) is a general problem solving method with a simple and appealing definition (Kim and Shin 2000) that emphasizes the findings of appropriate past experiences as a solution to new problems. The central tasks that the CBR method deals with are the identification of the current problem situation, finding a past case similar to the new one, and then using that case to suggest a solution to the current problem, evaluate the proposed solution, and update the system by learning from this experience (Riesbeck and Schank 1989, Slade 1991, Kolodner 1993, Aamodt and Plaza 1994). Recent successful applications of the CBR method point out the following: engineering applications including software estimation (Finnie *et al*. 1997), development of document retrieval systems (Watson *et al*. 1997), identifying failure mechanisms (Liao *et al*. 2000), and due date assignment (Chang *et al*. 2001, Chiu *et al*. 2003); business applications including bond rating (Shin and Han 1999, 2001, Kim and Han 2001) and bankruptcy prediction (Bryant 1997, Jo *et al*. 1997).

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

International Journal of Production Research
ISSN 0020–7543 print/ISSN 1366–588X online © 2006 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00207540500168238

The most basic problem in CBR is the retrieval and selection of relevant cases, since the remaining operations of adaptation and evaluation will succeed only if the past cases are relevant (Lopez de Mantaras 2001). The retrieval of relevant cases is closely related and dependent upon the indexing approach used. The indexes organize and label cases in the case base with the aim of deciding under what circumstances the cases may be useful. Indexing the cases in the memory means that the computer does not have to search each case stored in the case base for each case selection, because that would be considerably slower. Several case indexing approaches have been proposed in previous years. Among these approaches, the inductive learning-indexing (IL-indexing) approaches are widely used (e.g. in Cognitive system's ReMind) and commonly used variants of the ID3 algorithm used for rule induction (Watson and Marir 1994). Through performing the IL-indexing approach the CBR system can emphasize feature-value pairs for retrieving more relevant cases by inducting a tree-shaped structure and can make effective use of statistical measures to eliminate noise for a case retrieval. However, up to now there is no IL-indexing related approach suited to index the class of cases that takes on a numerical value. Therefore, in this study a modified form of IL-indexing approach is proposed for numeric prediction and is called the tree-indexing approach. It inherits the advantages and characteristics of the IL-indexing approach, in order to support CBR for predicting the numerical values efficiently and precisely.

In order to evaluate the effectiveness of the tree-indexing approach for CBR in numeric prediction, we have conducted experiments with conventional CBR and on CBR with the tree-indexing approach (T-CBR) on three real world problems from the UCI repository. This study also applies the T-CBR to solve the due date assignment problem in a dynamic job shop environment in order to investigate whether T-CBR's expected benefits can be observed in practice. The experiment's results show that our proposed T-CBR is significantly better than the existing prediction methods in reducing the prediction error. In summary, the contribution of this study is to develop the tree-indexing approach for improving the performances of CBR in numeric prediction, e.g. due date assignment, price prediction, estimation of the percentage of body fat, etc.

The remainder of this paper is organized as follows. In section 2 the related works on case-based reasoning and due date assignment are summarized. In section 3 the T-CBR method is introduced. Simulation experiments conducted to compare performances are described in section 4. Finally, in the last section conclusions are drawn and suggestions are made for future study.

2. Related work

In order to describe the process of developing the tree-indexing approach to CBR for predicting the continuous numeric values (e.g. due date assignment), it will be helpful to first discuss the following two areas as background: case-based reasoning and due date assignment.

2.1 *Case-based reasoning*

Case-based reasoning is a problem-solving technique in which past cases and experiences are used to find a solution to particular problems (Shin and Han 2001). The overview of the case-based reasoning process is shown in figure 1. The most basic problem in CBR is the retrieval and selection of relevant cases, since the remaining operations of adaptation and evaluation will succeed only if the past cases are the relevant ones (Lopez de Mantaras 2001). The retrieval of relevant cases is closely related to and dependent upon the indexing approach used. The case indexing problem has two parts. First is that of assigning labels to cases at the time they are entered into the case library to ensure that they can be retrieved when needed. Second is the problem of organizing cases so that the search through the case library can be done efficiently and accurately (Kolodner 1993). Given a description of a problem, the retrieval algorithm relies on the indices and the organization of the memory to direct the search to potentially useful cases.

There are five approaches for case indexing (Watson and Marir 1994): checklist-based indexing, difference-based indexing, similarity and explanation-based generalization methods, inductive learning methods, and explanation-based techniques. Among these approaches, the inductive learning-indexing (IL-indexing) approaches are widely used (e.g. in Cognitive system's ReMind) and commonly used variants of the ID3 algorithm used for rule induction (Watson and Marir 1994).

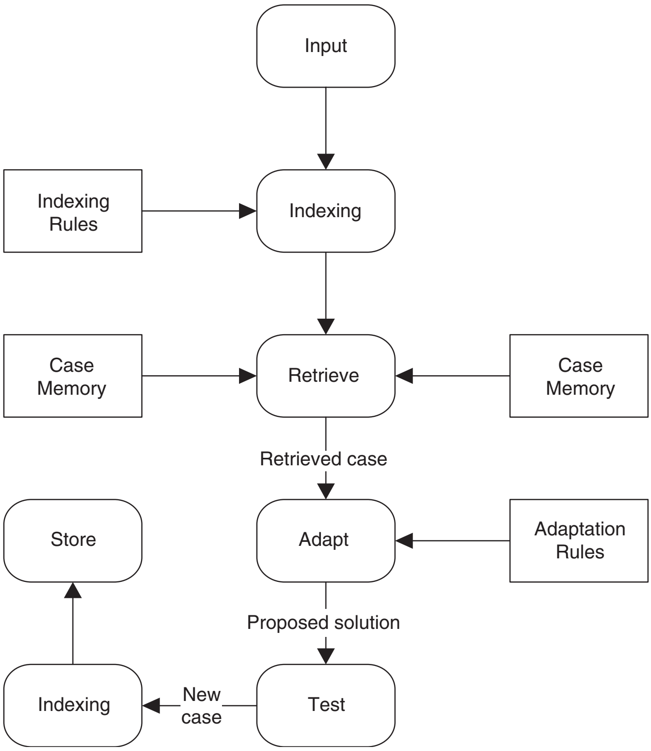

Figure 1. Overview of the case-based reasoning process (Riesbeck and Schank 1989).

The IL-indexing approach clusters cases that are similar to one another and figures out which category best matches the new situation. It then selects the most similar items in that category and adapts it as the new solution. This means that the computer does not have to search each case stored in case base for case retrieval, which would be much slower. Specially, the IL-indexing approach for case indexing is to build a tree-shaped structure. The traditional induction algorithms such as ID3 and C4.5 determine which features do the best job in discriminating cases and then generate the tree-shaped classification structure to organize and index the cases in the case memory. Because this kind of induction algorithm is designed for predicting categories rather than numerical quantities, it requires the target classes to be a nominal attribute when producing a decision tree. When it comes to indexing the cases in CBR for predicting numerical quantities, as with the due date assignment problem in our paper, the inducting manner needs to be modified. Therefore, a modified form of IL-indexing approach is proposed as an assistant to support CBR in indexing cases when predicting a 'target class' that takes on a numerical value.

2.2 *Due date assignment*

With the current emphasis on the just-in-time (JIT) production philosophy, it is crucial to meet the target job due date. Assigning exact due dates, and timely delivering of goods to the customer will enhance customer's satisfaction as well as provide a competitive advantage. In order to satisfy the customer's delivery performance, shop floor managers must accurately predict the due dates of the jobs under controllable conditions (Udo 1993).

To date, many regression-based due date assignment methods have been proposed, and the advantages of these classical approaches are easy to comprehend and practice. Initially, researchers examined due date assignment methods that considered only job characteristics in quoting the due date. Some of this kind of due date assignment method investigations were conducted 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), and (iv) random allowance (RAN). He finds that the methods which utilize the job information perform better than the others. Later, Eilon and Chowdhury (1976) compared the following two approaches of due date setting based on the information of job characteristics (e.g. TWK, SLK, NOP), and of job characteristics and shop status (e.g. jobs in system (JIS), jobs in queue (JIQ)). Based on their results, the later due date assignment methods perform better than the former one when used in conjunction with due date oriented dispatching rules. Vig and Dooley (1991) presented two new dynamic due date assignment rules which utilize shop congestion information: operation flow time sampling (OFS) and congestion and operation flow time sampling (COFS). Both rules estimate the job flow time based on sampling of recently completed jobs. The results clearly indicate that flow times from recently completed jobs provide very useful information for establishing effective due dates in a job shop environment. Gee and Smith (1993) proposed an iterative procedure for estimating flow times when due date oriented dispatching rules are used. Their results indicate that the global rule that utilizes both job and shop related information yields better estimation and that the quality of flow time estimation is improved by the iterative procedure. Smith *et al.* (1995) updated the regression coefficients of the regression equation that was based on the information of critical path length after completing 200 orders, in order to improve the prediction capability of the due date. Recently, researchers have studied multi-stage regression models. Among them, a new regression-based flow time estimation method that utilizes detailed job, shop and route information was developed to predict the flow time of each job operation (Sabuncuoglu and Comlekci 2002). The results of their study indicated that estimating flow times for each operation (i.e. operation by operation) is a better approach than the traditional job-based estimation and that the use of detailed information in estimating flow times provides significant improvement in the system performance over the other methods that utilize more aggregate information. In another study, the operation flow time prediction equation is proposed as a quadratic model, which is based on the information of the processing time of the operations and the average utilization of the machines (Veral 2001). This prediction model improves the accuracy of manufacturing lead time predictions over the TWK method in the simulated environment. The above regression-based prediction models are grouped as part of the conventional due date assignment methods. The regression-based due date assignment method is probably the most familiar technique, which has a considerable advantage, but its disadvantage is that the optimal regression model (i.e. linear, nonlinear) is very difficult to achieve.

In recent years, many artificial intelligent and machine learning tools have been used for decision support and generating forecasts. Philipoom *et al.* (1994, 1997) considered a new procedure for internally setting due dates, namely, neural network prediction, and found improved due date setting performance with neural networks as the methodologies of choice. Chang *et al.* (2001) explored case-based reasoning in the due date assignment problem of the wafer fabrication factory, the experimental results have shown that the CBR approach is very effective and comparable with a neural network approach. Chiu *et al.* (2003) proposed a CBR approach that employs the k nearest neighbour concept with dynamic feature weights and non-linear similarity functions. The test results of their approach have shown that it can more accurately predict order due dates than other approaches (neural networks, JIQ, TWK, NOP). Our purpose here is to begin the development of a novel case indexing approach for CBR for numeric prediction and attempt to see whether the due date setting performance of CBR with this novel case indexing approach can outperform conventional regression-based due date assignment methods that are commonly used in research and in practice, neural networks, and regular CBR.

3. Development and evaluation of T-CBR

This section discusses in detail the proposed CBR with tree-indexing approach for predicting continuous numeric values. The section is divided into three subsections, namely, tree-indexing approach, case retrieval and adaptation and applying T-CBR to real world problems from the UCI collection.

3.1 *Tree-indexing approach*

In this study, we propose the tree-indexing approach for CBR in indexing and retrieving of cases, in order to predict the numerical values. The novel approach derives the general domain knowledge from the case base for organizing the cases in the memory, so that it may support and enhance the retrieval of relevant cases to the problem. This general domain knowledge, which is expressed in a tree-shaped structure, means the identification of which feature-value pairs have higher predictive capability for problem solving.

CBR with tree-indexing approach (T-CBR) utilizes the strength of CBR as a prediction tool, and the tree-indexing approach as assistance in indexing and retrieving cases. When the query case arrives, the T-CBR works as follows: first, we apply the tree-shaped structure to retrieve a class of cases that are similar to the new one. On this subset of cases, the k nearest neighbour algorithm (k-NN) is performed to find the k best matches. After case retrieval, the solution is adapted based on the k best matches by adaptation rule. Among these procedures, the applications of the tree-shaped structure and the k-NN denote the utilization of general domain knowledge and case-specific knowledge in the case retrieval process.

In order to retrieve the relevant cases efficiently, the cases in the case base must be classified according to the distribution of their target class, and a classification method is then adopted to find the number of classes. Owing to the target class taking on a continuous numeric value, the traditional induction algorithms (ID3, C4.5) are not suited to classify the case base for case indexing. In this study, the basic idea behind building the index of cases for predicting the numeric value is quite straightforward, and is inspired by the concept of the decision trees algorithm. The new case indexing approach for the CBR is called the tree-indexing approach, and it involves two distinct processes.

First, a tree induction algorithm is used to build a tree for classifying the cases. Instead of maximizing the information gain at each interior node, a splitting criterion is used that maximizes the difference of the centroid values in the target class along each branch. The splitting criterion is based on calculating the expected difference in centroid values as a result of testing each feature-value pair at that node. The difference of centroid values ($Diff_{\text{centroids}}$) is calculated by equation (1)

$$Diff_{\text{centroids}} = abs(E(T_1) - E(T_2)) \tag{1}$$

where T_i and $E(T_i)$ denote the subset of cases that have the ith outcome of the potential test, and the mean of the class values of the cases in T_i respectively. Figure 2 gives a pseudo-code for the first stage in the tree-indexing approach. The main part is creating a tree by successively splitting nodes, performed by split. The tree-indexing approach does not split a node if the Max $Diff_{\text{centroids}}$ of the examples at the node is less than 5% of the mean of the class values of the entire case base of examples. The node data structure contains: a type flag indicating whether it is a leaf, pointers to the left and the right child, the set of examples that reach that node, and the feature-value pair that is used for splitting at that node. The E function called at the beginning of the main program, and again at the split calculates the mean of the class values of a set of examples. In split, size of returns the number of elements in a set. The $Diff_{\text{centroids}}$ is calculated according to equation (1). Second, after the tree is built, each case in the case base will be

```
Tree-growing (examples)
{
  MV = E (examples)
  T = sizeof (examples)
  for each k-valued enumerated attribute
    convert into k-1 synthetic binary attributes
  root = new_node
  root.examples = examples
  split (root)
}

split (node)
{
  if node.type <> LEAF
    for each continuous and binary attribute
      for all possible split positions
        if sizeof (node.left) > $P$ * T and size of (node.right) > $P$ * T
          calculate the feature-value pair's $Diff_{centroids}$
        else
          $Diff_{centroids}$ = 0
    if $MaxDiff_{centroids}$ < 0.05 * MV
      node.type = LEAF
    else
      node.splitpoint = attribute-value pair with Max $Diff_{centroids}$
      split (node.left)
      split (node.right)
}
```

Figure 2. Pseudo-code for first stage in the tree-indexing approach.

classified into a leaf. Each leaf in the tree denotes a unique class, which means a particular index. Each case in the case base is indexed according to the leaf into which the case falls. The success of the inductive indexing approach depends largely upon the appropriateness of decision trees for case retrieval (Kolodner 1993). To find an optimal or near optimal tree, three different branch sizes (P) for the tree growing are applied (= 5%, 15%, 25%). The branch size denotes the minimal number of cases a branch should have.

The reason why the tree-indexing approach is used to index the cases for numeric prediction is the fact that the tree-shaped structure divides the original case base into the numbers of distinct subsets of cases. Therefore, the retrieval time is kept as small as possible, and the cases that are retrieved are those that are the best for the query case.

3.2 *Case retrieval and adaptation*

After the tree-shaped structure is built using the above procedures on a database of previous cases; when the new case arrives, our T-CBR works as follows: first, we apply the tree-shaped structure to retrieve a subset of cases that are similar to the case in question. On this subset of cases, we perform nearest-neighbour algorithm (k-NN) to find the k best matches. This allows the examiner to determine the most similar cases to the current situation, and to choose the most probable k value in each subset of cases. In the implementation of CBR, the number of retrieved cases (k) will affect the smoothness of the forecasts. In general, large number of retrieved cases leads to smooth forecasts, and small number of retrieved cases leads to sharply varying forecasts. Therefore, for each subset of cases in the memory, the experiments were designed to forecast the target value of the validating examples, by varying the number of retrieved similar cases from 1 to 50. For each subset of cases, the optimal k values are determined by the exhaustive enumerations. Once the optimal number of retrieved cases is determined, the T-CBR is completely modelled.

In the implementation of k-NN, the similarity between an old case (say case$_p$) and a given query case (say case$_q$) can be measured using the standard Euclidean distance metric as in equation (2),

$$DIS_{\text{case}_p,\text{case}_q} = \sqrt{\sum_{i=1}^{n} W_i \times (C_i - T_i)^2} \tag{2}$$

where $C_i \in \text{case}_p$ and $T_i \in \text{case}_q$. In T-CBR, the feature weight W_i is equal to 1. After calculating the similarity between a query case (case$_q$) and each case in the retrieved class, the k cases that are similar to the query case are identified. The expected target value (TV_t) of the query case is derived by Jo *et al.* (1997) and is obtained by equation (3),

$$E\left(TV_t \middle| \{S_{tb}\}_{b=1,\ldots,n}\right) = \sum_{b=1}^{n} \left(\frac{S_{tb}}{\sum_{i=1}^{n} S_{ti}}\right) \times TV_b \tag{3}$$

where n, S_{tb}, and TV_b denote the number of cases selected to generate the forecast, the similarity between the query case t and the selected case b, and the flow time of selected case b for the subject application respectively. Among these, the similarity between the query case and the selected case is captured by inverting the distance between them. The framework of our proposed T-CBR is illustrated in figure 3.

3.3 *Applying T-CBR to real world problems from the UCI collection*

In order to compare the effectiveness of T-CBR for numeric prediction over the conventional CBR, this study performed experiments using three real world problems from the UCI collection (Blake *et al.* 1998). The data sets and their characteristics are listed in table 1. The second column in table 1 indicates the number of cases within the data set. The number of numerical features and number of categorical features are indicated in the third and fourth columns, respectively. The issue of *bodyfat* estimates the percentage of body fat as determined by underwater weighing and various body circumference measurements for 252 men. The *housing* data set concerns housing values in the suburbs of Boston. The lists in the *pwlinear* data set are generated by a piecewise linear function that is defined by Breiman *et al.* (1984). The generating function is the following.

$$F(X) = \begin{cases} 3X_2 + 2X_3 + X_4 + 3 + Z, & \text{if } X_1 = 1 \\ 3X_5 + 2X_6 + X_7 - 3 + Z, & \text{if } X_1 = -1 \end{cases}$$

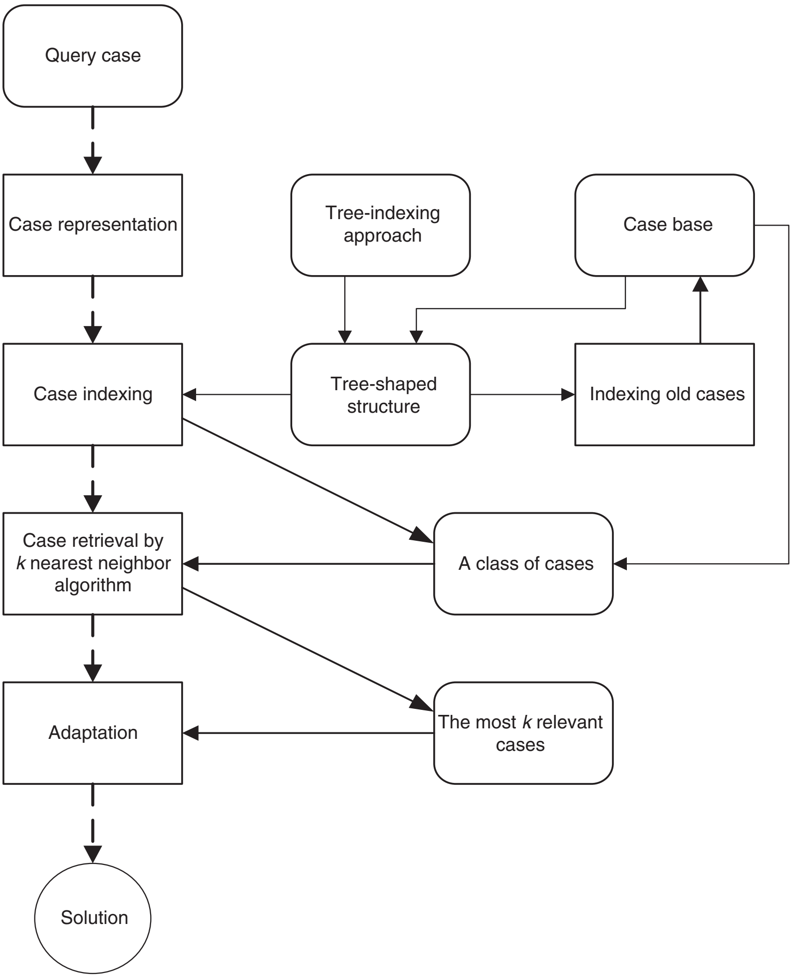

Figure 3. The framework of the T-CBR.

Table 1. Numeric class data sets.

Data set	Instances	Numeric	Nominal
Pwlinear	200	10	0
Bodyfat	252	14	0
Housing	506	12	1

Here, Z is a random Gaussian noise term. Features X_8, X_9, X_{10} have no bearing on class value. For each problem, four-fifths of the examples were randomly selected as training examples, and the remainder constituted the test samples. For determining the CBR parameters such as the k value in k-NN, four-fifths of the training examples were randomly selected as the case base, and the remaining ones were selected as the validating examples. The same case base was used in

Table 2. Performance comparison on real world problems from the UCI collection.

		Data set			
Methods	Branch size	*Pwlinear*	*Bodyfat*	*Housing*	Average
CBR	–	1.66	3.33	4.02	3.00
T-CBR	$P = 0.25$	***1.60***	3.42	3.74	2.92
	$P = 0.15$	2.07	2.91	4.33	3.10
	$P = 0.05$	1.65	***2.87***	***3.69***	2.74

determining the CBR and T-CBR parameters, and was used in testing. For each problem, the root mean square error (RMSE) made by the CBR and the three T-CBR models over the test samples are reported in table 2. Boldface and italic are used to indicate the best result for each problem. Among the models, T-CBR with $P = 0.05$ has the lowest level of RMSE on the *bodyfat* and *housing* data sets, T-CBR with $P = 0.25$ has the lowest level of RMSE on the *pwlinear* data set. Based on the results, we can conclude that the tree-indexing approach employing an optimal or near-optimal level of general domain knowledge is effective, enhancing the overall prediction performance of the case-based system for the application domain. Our experiments indicate that T-CBR with specific P performs much better than the conventional CBR in all of the data sets. This result also underlines the necessity of optimising branch sises applied in a case-based retrieval.

4. Due date assignment in T-CBR

Because the shop conditions when the new job arrives are probably similar to those of previously arrived jobs, the CBR method provides a suitable means for solving the due date assignment problem. We have applied the T-CBR for solving the due date assignment problem in a dynamic job shop environment, in order to investigate whether T-CBR expected benefits are observed in practice. The section is divided into five subsections, namely, the job shop model, data collection, feature selection, due date assignment methods, and results of the experiments.

4.1 *The job shop model*

The strength of T-CBR is based on the combined utilization of general domain knowledge and case-specific knowledge. In the field of due date assignment, the general domain knowledge, such as job scheduling knowledge regarding due date assignment, relies on making associations along generalized relationship principles between the condition of the shop when the job arrives and the actual flow time of the job. For this experiment, a suitable shop model needs to be defined. This research used a 10 × 10 benchmark problem from Lawrence (1984). Table 3 provides the data for the problem using the following structure: machine, processing time. The probability of each product being chosen to be released into the shop is equal. Job inter-arrival times were also selected from a negative exponential

Table 3. 10 × 10 job shop problem (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

distribution, but with a mean of 76.5. This resulted in a shop utilization of 90%, which represents a heavy shop load. As mentioned, the 'shortest processing time' dispatching rule (SPT) was used in this study for two reasons (Philipoom *et al*. 1994). First, it has been shown in several studies that when a shop is highly congested, SPT outperforms due-date-based dispatching rules for mean tardiness. Second, assigning internally set due dates in a shop is much more difficult for a manager using SPT as a dispatching rule than it is for a manager using either a minimum slack or a first-come, first-served dispatching alternative (Ragatz and Mabert 1984). Therefore, SPT is the dispatching rule that is being used in the present study. The virtual job shop was built on a personal computer with a Pentium III 700 processor using the eM-Plant 4.6, a simulation package developed by Tecnomatix Technologies Ltd.

4.2 *Data collection*

This study collected a large amount of data using a simulation experiment in a virtual job shop. It is necessary to guarantee statistical independence among the cases before the test is performed. To ensure this, once the simulation reached the steady state, only one in every 50 outputs from the shop simulation was randomly selected to be included in the sample of 1000 jobs as the data set. The warm-up period for the shop was the time interval from the start of the simulation to the completion of the first 10 000 jobs. In particular, for each collected job, two general job characteristics data were obtained, as well as forty-two shop status data, resulting in 44 characteristics per collected job. As a result, all features were reflective of the condition of the shop at the instant the job entered the shop. In addition, the actual flow time through the shop was observed, and from this the complete data was determined (see table 4). They are a mix of numeric and categorical data. Before using them, the numerical features are normalized to lie in a fixed range, say from zero to one for finding the distance (equation (1)); if any categorical features are selected, then dummy variables will be used as substitutes.

Table 4. Complete data for each collected job.

Factor	Information
Job characteristic	
Job type	The type of job
TW	Sum of processing times for job *i*
Shop status	
M1QL, ..., M10QL	Sum of the jobs presently in queue on machine 1, ..., 10
M1WL, ..., M10WL	Sum of the remaining processing time on the machine 1, ..., 10 for all the jobs in the shop
NJ1, ..., NJ10	Work in process of job 1, ..., 10 in the shop
J1R, ..., J10R	Average flow time obtained from the three most recently completed jobs of job 1, ..., 10
SRT	Sum of the remaining processing time for all jobs in the shop
WIP	Work in process in the shop
Target continuous-class	
FT	Actual flowtime in the system of job *i*

Table 5. Case representation.

Factor	Information
Job characteristic	
Job Type	The type of job
TW	Sum of processing times for job *i*
Shop status	
M2QL	Sum of the jobs presently in queue on machine 2
M7WL, M10WL	Sum of the remaining processing time on the machine 7, 10 for all the jobs in the shop
NJ9	Work in process of job 9 in the shop
SRT	Sum of the remaining processing time for all jobs in the shop
Target continuous-class	
FT	Actual flowtime in the system of job *i*

4.3 *Feature selection*

As discussed earlier in section 4.2, many features were gathered for the collected jobs. Some features influence the flow time, while others do not. This experiment acquired seven features, that is to say, two job characteristics and five job statuses by means of screening with a stepwise regression procedure for the data set, in order to select statistically significant input features. Once the influential features were identified, the case of each job in the data set was represented by these features, and by their actual flow times. The case of each job is represented by its influential features and flow time that are listed in table 5. For example, when a new job arrives with *Job Type* 8, *M2QL* 0, *M7WL* 557, *M10WL* 404, *TW* 366, *SRT* 4298, *NJ9* 4, and actual flow time 756, the case of this job can be represented using the row vector [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 557, 404, 366, 4298, 4756].

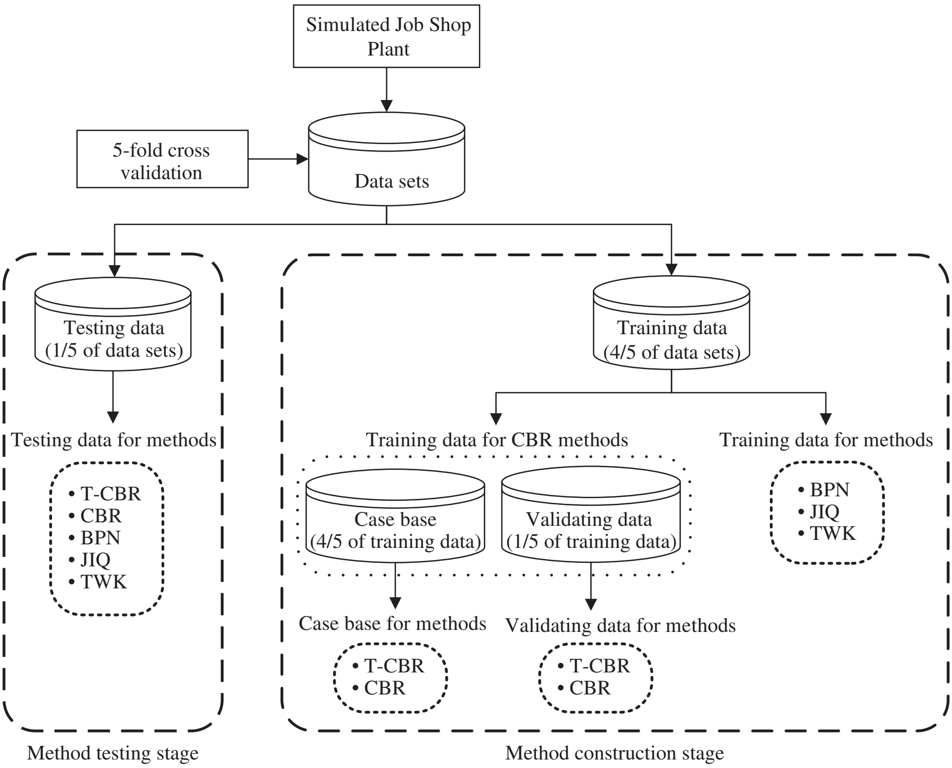
Simulated Job Shop Plant
5-fold cross validation
Data sets
Testing data (1/5 of data sets)
Testing data for methods
• T-CBR
• CBR
• BPN
• JIQ
• TWK
Training data (4/5 of data sets)
Training data for CBR methods
Training data for methods
Case base (4/5 of training data)
Validating data (1/5 of training data)
• BPN
• JIQ
• TWK
Case base for methods
Validating data for methods
• T-CBR
• CBR
• T-CBR
• CBR
Method testing stage
Method construction stage

Figure 4. Data distribution of method construction stage and testing stage.

4.4 *Due date assignment methods*

For comparison with the proposed T-CBR, conventional case-based reasoning (CBR), back-propagation neural networks (BPN), jobs in queue (JIQ), and total work content (TWK) due date assignment methods were chosen. Both the T-CBR and CBR employed the same similarity metric with equal weights. The CBR retrieved the relevant cases from the overall case base, but the T-CBR retrieved the relevant cases from the specific subset of cases that had the same index with the query case. By comparing the performance of T-CBR with the performance of CBR, it ensured a level playing field in which the two types of system used the same information for evaluating the effectiveness of the tree-indexing approach. The BPN is the most commonly used technique in forecast problems; it uses the gradient steepest descent method to minimise the total square error of the output computed by the net. The JIQ and TWK are grouped into the conventional due date assignment methods that are modelled by regression analysis.

4.5 *Experimental results*

As illustrated in figure 4, all methods were tested with a five-fold cross validation method. For determining the parameters in T-CBR and CBR, such as the *k* values

Table 6. The RMSE for each method.

DDA methods		RMSE
Machine learning methods	T-CBR	***478.60***
	CBR	539.09
	BPN	529.31
Conventional methods	JIQ	578.00
	TWK	605.96

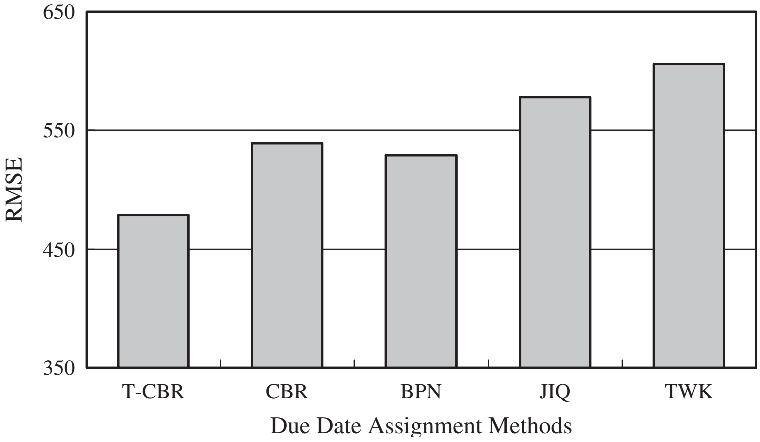

Figure 5. The RMSE for each method.

in *k*-NN, four fifths of the training data for each five-fold cross validation was randomly selected as the case base, and the remaining ones were used as the validating data for parameter setting of CBR systems. CBR, BPN, JIQ, and TWK were established by using the same training data for benchmarking as for the T-CBR. The RMSE is used consistently to measure the performances of all the due date assignment methods.

For each fold, the T-CBR was designed to forecast the job due date by varying the parameter *P* from 5% to 25%. The best performance is illustrated in table 6 and figure 5. Simultaneously, the RMSE made by other tested method over the test samples is reported in table 6 and figure 5. The best result for the due date assignment problem is highlighted in bold face and italic. The results in table 6 demonstrate that all machine learning methods outperform conventional methods with respect to RMSE. Among these machine learning tools, the T-CBR appears to be the best. The RMSE in T-CBR reduced by 11.22% compared to the CBR, reduced by 9.58% compared to that of the BPN, and reduced by 17.19–21.01% over that of the conventional methods. Besides, the case base in T-CBR is divided into several subsets and so the efficiency of the case retrieval can of course be better than that of CBR. Based on these results, the tree-indexing approach is an effective method for CBR for improving the performance of the numeric prediction.

5. Conclusions

In this study, we presented a novel case indexing approach of CBR for numeric prediction, referred to as the tree-indexing approach, which improved the effectiveness and the efficiency of conventional CBR by inducing the tree-shaped structure to assist in the indexing and retrieval of cases. The advantages of tree-indexing approach are:

1. It can index the cases at the retrieval stage for predicting the class of cases that takes on a numerical value, rather than a discrete category into which an example falls.
2. The tree-shaped structure divides the original case base into a number of distinct subsets of cases, and so the retrieval time is kept as small as possible.
3. It can lessen the effect of the irrelevant cases that may be retrieved by inducting a tree-shaped structure to identify small sets of highly predictive feature-value pairs.

In order to investigate whether the expected benefits of the tree-indexing approach are observed in practice, we compared T-CBR with the conventional CBR, which are similar to T-CBR but without a tree-indexing part, in three real-world problems from the UCI repository. Our experiments showed that CBR with the tree-indexing approach outperforms the conventional CBR with respect to RMSE, especially in setting an appropriate branch size (P) for the tree growing procedure.

Furthermore, this study applied the T-CBR to forecast the job due date in a dynamic job shop environment. In the field of due date assignment, the general domain knowledge derived by the tree-indexing approach, namely, job scheduling knowledge regarding due date assignment, relies on making associations along a generalized relationship between the condition of the shop when the job arrives and the actual flow time of the job, and is represented as a concrete tree-shaped structure. For comparison purpose, conventional CBR, BPN, JIQ, and TWK due date assignment methods were used to construct the prediction model as well. The results of these experiments indicated that the T-CBR exhibits a performance that is superior to that of the conventional CBR and other methods for the due date assignment problem. In summary, the tree-indexing approach clearly can improve the effectiveness and efficiency of CBR.

The limitation of tree-indexing approach is that the effectiveness and efficiency of CBR can not be improved when applied it to index a very homogenous data set. For this kind of data set, because the tree-indexing approach is unable to find any clusters from it, the mean retrieval and adaptation time of the cases picked up by T-CBR will be equal to the cases picked up by the CBR, and the cases picked up by T-CBR and CBR are the same. However, in our study the results of the four data sets (*pwlinear*, *housing*, *bodyfat*, and *jobshop*), whereby the T-CBR always performs faster and more precise than the CBR.

The determination of the branch size in the tree-growing procedure has an impact on the performance of the T-CBR. We intend to find a general method to determine the branch size in future research. Future studies might want to focus on investigating whether further improvement can be made by a better design of the T-CBR. In addition, an obvious area for future research is to carry out similar tests for a more realistic shop setting for standard products. Lastly, using hierarchical clustering technique to develop the tree is worth issue in future research.

References

Aamodt, A. and Plaza, E., Case-based reasoning: foundational issues, methodological variations and system approaches. *AI Communications*, 1994, **7**(1), 39–59.

Blake, C.L. and Merz, C.J., *UCI Repository of Machine Learning Databases* [http://www.ics.uci.edu/~mlearn/MLRepository.html], 1998 (University of California, Department of Information and Computer Science: Irvine, CA).

Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J., *Classification and Regression Trees*, 1984 (Wadsworth: Belmont, California).

Bryant, S.M., A case-based reasoning approach to bankruptcy prediction modelling. *Int. J. Intel. Syst. in Acc., Finance & Manage.*, 1997, **6**(3), 195–214.

Chang, P.-C., Hsieh, J.-C. and Liao, T.W., A case-based reasoning approach for due-date assignment in a wafer fabrication factory, in *Proceedings of the 4th International Conference on Case-Based Reasoning*, Springer-Verlag, Berlin, Germany, 2001, pp. 648–659.

Chiu, C.-C., Chang, P.-C. and Chiu, N.-H., A case-based expert support system for due-date assignment in a wafer fabrication factory. *J. Intel. Manuf.*, 2003, **14**(3–4), 287–296.

Conway, R.W., Priority dispatching and job lateness in a job shop. *J. Indust. Eng.*, 1965, **16**(4), 228–237.

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

Finnie, G.R., Witting, G.E. and Desharnais, J., Estimating software development effort with case-based reasoning, in *Proceedings of the 2nd International Conference on Case-Based Reasoning*, Springer Verlag, Berlin, Germany, 1997, pp. 13–22.

Gee, E.S. and Smith, C.H., Selecting allowance policies for improved job shop performance. *Int. J. Prod. Res.*, 1993, **31**(8), 1839–1852.

Jo, H., Han, I. and Lee, H., Bankruptcy prediction using case-based reasoning, neural networks and discriminant analysis. *Exp. Syst. with Applications*, 1997, **13**(2), 97–108.

Kim, K.S. and Han, I., The cluster-indexing method for case-based reasoning using self organising maps and learning vector quantisation for bond rating cases. *Exp. Syst. with Applications*, 2001, **21**(3), 147–156.

Kim, S.H. and Shin, S.W., Identifying the impact of decision variables for nonlinear classification tasks. *Exp. Syst. with Applications*, 2000, **18**(3), 201–214.

Kolodner, J., *Case-Based Reasoning*, 1993 (Morgan Kaufmann: San Mateo, CA).

Lawrence, S., *Resource Constrained Project Scheduling: an Experimental Investigation of Heuristics Scheduling Techniques*, 1984 (Graduate School of Industrial Administration: Carnegie Mellon University, Pittsburgh).

Liao, T.W., Zhang, Z.M. and Mount, C.R., A case-based reasoning system for identifying failure mechanisms. *Eng. Applic. Artif. Intel.*, 2000, **13**(2), 199–213.

Lopez de Mantaras, R., *Case-based Reasoning. Lecture Notes in Computer Science 2049*, 2001, pp. 127–145 (Springer-Verlag: Heidelberg).

Philipoom, P.R., Rees, L.P. and Wiegmann, L., Using artificial neural networks to determine internally-set due date assignment for shop scheduling. *Decision Sciences*, 1994, **25**(5/6), 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 Research Logistics*, 1997, **44**(1), 21–46.

Ragatz, G.L. and Mabert, V.A., A simulation analysis of due date assignment rules. *J. Op. Manage.*, 1984, **5**(1), 27–39.

Riesbeck, C.K. and Schank, R.C., *Inside Case-Based Reasoning*, 1989 (Lawrence Erlbaum Associates: Hillsdale, NJ).

Sabuncuoglu, I. and Comlekci, A., Operation-based flow time estimation in a dynamic job shop. *Omega*, 2002, **30**(6), 423–442.

Shin, K.S. and Han, I., Case-based reasoning supported by genetic algorithms for corporate bond rating. *Exp. Syst. with Applications*, 1999, **16**(2), 85–95.

Shin, K.S. and Han, I., A case-based approach using inductive indexing for corporate bond rating. *Decision Support Systems*, 2001, **32**(1), 41–52.

Slade S., 1991, Case-based reasoning: a research paradigm. *AI Magazine*, **12**(1), 42–55.

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**(9), 2375–2385.

Udo, G., An investigation of due date assignment using workload information of a dynamic shop. *Int. J. Prod. Econ.*, 1993, **29**(1), 89–101.

Veral, E.A., Computer simulation of due-date setting in multi-machine job shops. *Computers & Indust. Eng.*, 2001, **41**(1), 77–94.

Vig, M.M. and Dooley, K.J., Dynamic rules for due date assignment. *Int. J. Prod. Res.*, 1991, **29**(7), 1361–1377.

Watson, I. and Marir, F., Case-based reasoning: a review. *The Knowledge Engineering Review*, 1994, **9**(4), 355–381.

Watson, I. and Watson, H., CAIRN: a case-based document retrieval system, in *Proceedings of the 3rd United Kingdom Case-Based Reasoning Workshop*, University of Manchester, edited by N. Filer and I. Watson, 1997.