Computer-Aided Civil and Infrastructure Engineering **13** (1998) 143–150

SHORT CONTRIBUTION

Case-Based Reasoning in Construction Management

Nie-Jia Yau* & Jyh-Bin Yang

Department of Civil Engineering, National Central University, Chungli 320, Taiwan, Republic of China

Abstract: *Case-based reasoning* (*CBR*) *is a highly effective technique in the artificial intelligence* (*AI*) *domain capable of solving or providing suggestions for the current problem by storing and retrieving results of previous cases. In the experience-oriented construction industry, knowledge and assessments of previous experience are critical to resolving problems that may reoccur. Applying CBR to construction management overcomes some drawbacks of other AI technologies and adheres to computer application needs in industry. CBR provides an alternative in solving experience-oriented problems. This article not only presents a typical CBR approach but also investigates its potential applications in construction management. An example shows how CBR can be used to estimate construction duration and costs of building construction projects at the preliminary design stage.*

1 INTRODUCTION

The construction industry is experience-oriented. In construction management, experienced expertise is crucial to solving problems and providing suggestions. In this domain, some problem solving has used artificial intelligence (AI) technologies extensively, particularly rule-based expert systems (ESs) and neural networks (NNs). However, these techniques have various drawbacks in applications: ESs lack the capability to learn by themselves, and NNs deal primarily with problems that could be presented in numerical figures. In addition, since each construction project is unique, and since human experts derive knowledge directly from the projects they have experienced, technologies other than ESs or NNs must be investigated to yield a better solution.

In recent years, case-based reasoning (CBR) has grown from a rather specific and isolated research area to a field receiving widespread interest in the AI domain.[1] This case-oriented and analogy-based technique can avoid the drawbacks of ESs and NNs by storing and retrieving previous solutions (experience) to solve (or provide suggestions for) a current problem. Therefore, exploring its applicability in the construction domain is a worthwhile task. This study employs a CBR approach to estimate construction duration and costs of a building construction project from 60 hypothetical projects with minimum input of each project's characteristics.

In the rest of this paper, Section 2 briefly introduces the concepts of CBR and discusses how it differs from ESs and NNs. Section 3 addresses potential CBR application areas in the construction management domain. Section 4 illustrates the CBR approach to estimate the duration and costs of building construction. And finally, concluding remarks are made.

2 BASIC CONCEPTS OF CASE-BASED REASONING

In CBR, experience is captured and organized as a set of historical cases stored in a case base. Similar cases are recalled from the case base to solve problems or provide suggestions. In the case base, a case is analyzed into several fields into which various values are filled. In a sense, the knowledge represented in the case base is quite similar to that stored in the "object-attribute-value" form, one of the methods of representing knowledge in expert systems (ESs). In such a representation, a *case* is like an *object*, and *features* of a case are just like *attributes* of the object. However, the values in the CBR case base are *real* values having occurred previously, not artificially made or induced. In other words, CBR is not an innovative approach in representing knowledge but instead uses the knowledge retrieved from similar old cases to solve or provide suggestions for solving new ones.

* To whom correspondence should be addressed.

Table 1
An illustrative example: Features of a construction project.

Problem description	
Feature	*Value*
Project name	ABC building
Owner	DEF, Inc.
Contractor	XYZ Co.
Site area	810 m^2
Structural system	Steel structure
Floors below ground	1
Floors above ground	4
Labor resource	Poor
Material resource	Available
Equipment resource	Limited
Solution description	
Feature	*Value*
Duration	250 days
Cost	$3,683,944

The stored case base is similar to a conventional database system with some particular features (fields). However, its retrieval is based on a situation's properties or the user's specific requirements. Moreover, full matching of features is not required, and a database administrator is not needed to formulate.[7] Thus a CBR approach is quite appropriate for the domains that are rich in experience but whose features are difficult to define.

2.1 Development of a CBR system

Developing a CBR system involves several typical processes, as shown in Fig. 1.[13] A case in the case base normally consists of two major portions: a problem description describing characteristics of the problem and a solution description describing results of or solutions to the problem. Each portion may contain various features to represent the contents of the case. An example is presented in Table 1.

After the case base is developed, case retrieval and case adaptation are the two fundamental processes in a CBR application (see Fig. 1). In the retrieval process, similar definitions of features determine which old cases are most similar to the new one. In other words, the CBR system examines the cases stored in the case base according to user-defined similarity for retrieval purposes. The case adaptation process involves a user-defined methodology that may directly capture, calculate, or manipulate the field values of the retrieved cases to generate a solution to the new problem. A new problem along with its generated solution can be validated and incorporated into the existing case base for enriching the case base. This validation process allows the CBR system to improve the system's performance over time by adding more cases. Such a process also allows for learning more experiences and avoiding previous mistakes.

2.2 Evaluation of the CBR approach

CBR, ESs, and NNs share a common goal of enhancing computer intelligence and making it more human-like.[5] Like CBR, ESs can be applied in experience-oriented and knowledge-intensive domains. However, the essence of an ES is a knowledge base represented primarily by transparent if-then rules, not a case base, so it is limited by the process of acquiring knowledge to establish the knowledge base. Moreover, in most cases, an ES cannot learn and has an extremely limited tolerance of incomplete input information when default values in the system are inadequate for the new problem. NNs are particularly appropriate for pattern-recognition problems, but they are based on numerical computations designed to adjust neuron weights in the net, thereby limiting the input and output to purely numerical figures. In NNs, the knowledge base in a trained net is deemed a "black box," since it is represented in a series of numerical vectors that cannot be understood easily by humans. Furthermore, NNs must have a large body of data sets for training (which is generally proportional to the number of data sets), and their problem-solving structure is normally defined by varying the number of layers and the number of neurons in each layer.

In CBR, neither knowledge acquisition nor numerical transformation is required, since the knowledge base is represented by previous cases. Notably, if new cases are incorporated, the case base can be easily updated. CBR retrieves cases and ranks them on the basis of the user-defined similarity function. As long as the input information generally fits into the similarity function, CBR does its retrieval. Therefore, CBR is more tolerant of incomplete information. Unfortunately, while CBR has advantages when only incomplete information is available, it may blindly generate a solution from the retrieved case based on such limited information. This drawback should be examined carefully when applying the CBR approach.

In CBR, each case is represented by a number of fields in various forms, such as numerical, logical, alphabetical, and strings. Thus a CBR application's input and output (as well as the case base itself) are more readable than that of a NN application. If previous cases are selected carefully and stored in the case base, a CBR application does not require training before it can be put to use, thereby saving much of the training time required by NN applications.

Developing a successful CBR application relies on the availability of previous cases and the definitions of the features that represent a case in the case base. To use the CBR approach, therefore, the characteristics of the pending problems must be clearly identified and expounded, based on the concept of similarity among cases. For domains where repre-

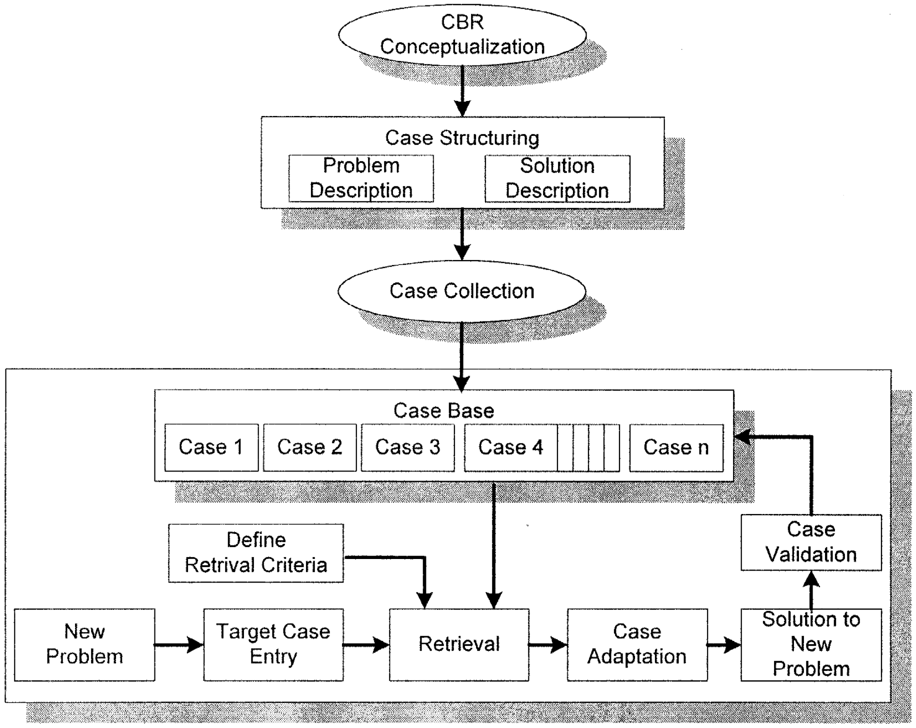

Fig. 1. General CBR development methodology.

senting knowledge by rules is extremely difficult, CBR makes knowledge representation more flexible. Moreover, it allows cases and solutions to be stored for later retrieval.

3 POTENTIAL CBR APPLICATIONS IN CONSTRUCTION MANAGEMENT

Each construction product is unique. However, construction processes and management skills are typically analogous. For this reason, several CBR systems have been proposed for the construction domain in recent years. For instance, ARCHIE[9] is an architectural design system, CADSYN[8] and DDIS[11] are structural design systems, and PAKAR[12] is a building defects diagnosing system. Additionally, the SEED[4] project provides computational support for the early design phase in which the CBR technique is incorporated to access previous design cases.

CBR can potentially be applied to various phases of a typical life cycle of construction projects, as explored in Fig. 2 and summarized in Table 2.[13] As described in the preceding sections, CBR requires adequate, expert definitions of features in the case base to make the case retrieval meaningful. For instance, estimating costs and duration in the preliminary design phase relies on calculations from several prominent features such as the number of floors and the type of structural system; these should be given by experts. Moreover, even with incomplete information (e.g., one or two key features), similar cases in the case base can still be retrieved.

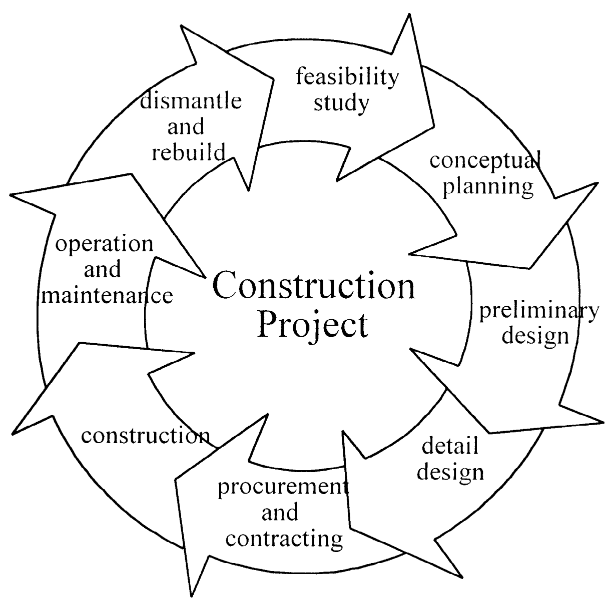

Fig. 2. Typical life cycle of construction projects.[13]

Table 2
Potential application areas of CBR in the life cycle of a construction project.[13]

Life-cycle phase	*Potential application areas*
Feasibility study	Risk analysis, feasibility analysis
Conceptual planning	Project requirements in design, costs and duration
Preliminary design	Design systems selection, conceptual time/cost estimates, selection of construction methods
Detail design	Architectural or systems design, design review criteria, time/cost predictions
Procurement and contracting	Selection of contractors, bidding price prediction, preparation of bidding documents
Construction	Site layout, schedule generation and control, budgeting and cost control, quality control, safety inspection, resources management
Operation and maintenance	Operation problems resolution
Dismantle and rebuild	Construction product assessments

CBR cannot be applied if the previous cases are unavailable. Unfortunately, detailed information regarding most construction projects, such as as-built schedules, is not well organized. Most of the time, only several key figures (e.g., final construction costs or project start/completion dates) are recorded. Under some circumstances, such as at the design stages, the CBR approach is not economically feasible unless the design information can be easily computerized. Therefore, the extent to which CBR can be implemented in construction management relies on the availability of construction information stored in a construction company.

4 ILLUSTRATION OF CBR APPLICATION: CONSTRUCTION DURATION AND COST ESTIMATION

For an example of CBR application, this study incorporates CBR techniques to estimate construction duration and costs for building construction at the preliminary design stage. At this stage, duration and cost are influential in determining which design is feasible or beneficial to the owner.

4.1 Case-base generation

In forming a case base for the CBR approach, accumulating actual construction project data is the most difficult task. To proceed with this approach, we managed to generate a hypothetical construction case base through a construction planning expert system, the Time/Cost Integrated System (TCIS).[14]

Running on KAPPA-PC,[6] a Windows-based expert system shell, TCIS is an object-oriented expert system for automatic scheduling and cost estimating of building construction at the preliminary design stage. To estimate duration and costs, TCIS incorporates rules from experienced construction experts and the mean cost data.[9] The input to TCIS is a set of 24 design parameters involving typical building construction (e.g., site area, number of floors above ground, number of floors below ground, and average floor height). The outputs of TCIS are a construction schedule and construction costs that are aggregated into various groups or levels. Although TCIS was not validated by real-life construction projects, its knowledge was solicited from domain experts, so the output of TCIS should fall into acceptable ranges at the preliminary design stage.

Since the purpose of this experimental application is to demonstrate CBR's feasibility in estimating construction costs and duration, hypothetical cases will not change the application processes. In the current study, the construction case base consists of 60 hypothetical projects generated by TCIS from random input of major features. Even though more cases can be put into the case base in a CBR application, we found that these 60 cases are sufficient to cover TCIS's ranges of output. Before generating the hypothetical projects, the following assumptions were made to simplify the construction projects' complexity:

1. The projects are limited to midrise concrete office buildings.
2. The foundation is pile with fixed size, there are no old buildings or trees are on the site, and the quality of lighting and plumbing fixtures is medium.
3. The interior partitioning is drywall, the exterior partition is concrete masonry wall, the floor covering is carpet, the heating system is a heating pump, and the fire protection system is sprinklers.
4. The output activities in the schedule are neglected except for the total project duration.
5. The output costs are labor, material, equipment, and total costs.

An example with input and output information is shown in Table 3.

Table 3
Features of an example project in the case base (bold-italic represents TCIS output information)

Feature	*Example value*
Project name	Pj01
Project start	1/1/1997
Project finish	12/31/1997
Site area (m^2)	405
No. of floors above ground	5
No. of floors below ground	1
Average floor height (m)	3.3
Slope of roof (in degrees)	0
Condition of resources	Shortage
Percentage of window (%)	20
Number of elevators	4
Percentage of pavement (%)	30
Percentage of landscaping (%)	20
Duration (in days)	***255***
Equipment cost (in dollars)	***191,900***
Material cost (in dollars)	***1,992,613***
Labor cost (in dollars)	***1,677,331***

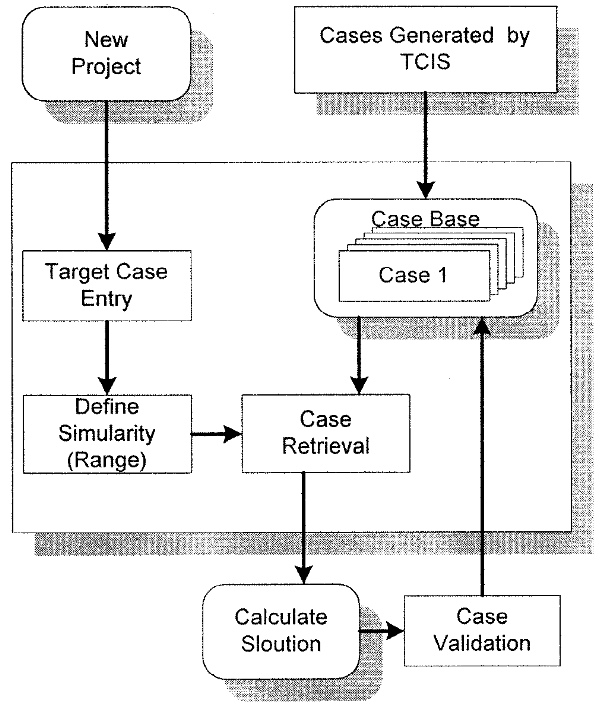

Fig. 3. System architecture of CBR-CURE.

Table 4. Characteristics of reasoning features for matching cases

Feature	*Entered range*	*Range for matching (similarity)*	*Weight for similarity evaluation*
Site area (m^2)	400–20200	±5	9
No. of floors above ground	2–10	Equal	7
No. of floors below ground	1–5	Equal	7
No. of elevators	2–10	Equal	5
Average floor height (m)	2.2–4.6	±10	3
Percentage of window (%)	0–100	±10	3
Percentage of pavement (%)	0–100	±10	3
Percentage of landscaping (%)	0–100	±10	3
Slope of roof (in degrees)	0–89	±10	1
Condition of resources	Shortage, medium, or plenty	Exact	Reasoning by rules

4.2 System architecture

In the following, a case-based reasoning system for building construction duration and cost estimation (CBR-CURE) is presented.[15] Fig. 3 depicts the system architecture.

CBR-CURE is built on a commercially available CBR software: ESTEEM, a Windows-based tool for developing decision-making applications by using previous problem-solving experiences (cases).[3] As mentioned in the preceding subsection, CBR-CURE uses 17 features to represent a case (see Table 3). In these 17 features, the project's name, as well as its start and finish dates, usually varies from project to project; the output portion of a case is the unknown for a new project. Thus the other 10 features are identified as the input information required for a new project (Table 4). They are entered through a user interface into the system to retrieve similar cases. Weights of each feature as well as its range of matching are subjectively assigned by the authors. The weight of "condition of resources" is determined by the rules in Fig. 4. These weights and ranges of matching can be changed if necessary. When a new project is entered as the target case, the system retrieves similar cases from the case base according to the range of matching in each feature.

In the matching process, a case is deemed similar and re-

Rule 1:
IF *TargetCase:condition_of_resources* **equal** *CurrentCase:condition_of_resources*
THEN *Similarity:condition_of_resourcesWeight = 5*

Rule 2:
IF *TargetCase:condition_of_resources* **not equal** *CurrentCase:condition_of_resources*
THEN *Similarity:condition_of_resourcesWeight = 1*

Fig. 4. Rules for determining weights of features.

trievable if its matching score is higher than the minimum score (criterion) input by the user. A case's score is calculated so that if a feature value of the case falls into the matching range (shown in Table 4), the feature weight is accounted for in the score, which will be finally normalized in a scale ranging from 0 to 100. That is, if the features of a case in the case base match every input feature, the case will have a score of 100. If the minimum score should be too high to retrieve a similar case from the case base, the minimum score must be lowered. In sum, the matching criteria are quite subjective, depending on the number of cases retrieved. Hence the retrieved cases are more adequate if an expert provides the matching criteria in the application domain.

After the similar cases are selected from the case base, the new project's intended construction duration and costs (target values) are computed by eq. (1), which is defined so that the score of each selected case determines the extent to which the case would contribute to the target value. In addition to the score, each selected case has a factor that adjusts the contribution of each selected case if certain conditions may affect the duration or costs. An example in which the adjusting factors are determined by rules is shown in Fig. 5. In this study, the values of the adjusting factors are assigned by the authors. The various adjusting factors provide a mechanism through which experts' opinions can be incorporated in calculating results from the retrieved cases.

$$\text{Target value (duration or cost)} = \frac{\sum(\text{score of selected case} \times \text{value (duration or cost) of selected case} \times \text{adjusting factor})}{\sum(\text{score of selected cases} \times \text{adjusting factor})} \quad (1)$$

The new project's input features (as well as the calculated construction duration and costs) can be added into the case base if these features are validated and accepted. This process can enrich the case base and further improve CBR-CURE's performance.

4.3 System test and results

The performance of CBR-CURE was assessed by three test projects. Each test project has four scenarios in which the number of input values was reduced. In other words, the feasibility of CBR to obtain results from various input was examined by reducing the numbers of input features from 10, 8, 6, to 4 (depicted as scenario 1, 2, 3, and 4, respectively). These results are summarized in Table 5. At the same time, construction duration and costs of these three test projects also were generated by TCIS (not real cases) for comparison. For the same project, the same cases were selected even if the number of input features was reduced from 10 to only 4 (see Table 5). When fewer features were input, fewer scores were obtained. Since the case base was randomly generated from TCIS, selecting the same cases demonstrates that CBR can retrieve some similar cases and produce a solution with incomplete input information. However, the premise of such an approach is that major features should be properly defined first. If this can be achieved for problems that require lengthy input, the CBR approach should be able to reduce the processing time to a minimum.

The results obtained by CBR-CURE are quite acceptable compared with the duration and costs generated by TCIS. The duration differences in the three test projects are less than 10%, while cost differences are significantly less than 20%. The higher cost differences could be due to low matching scores. However, this fact does not pose a serious problem, since previous studies have demonstrated that cost estimates should fall within the 16% range of the total building cost in conceptual cost estimating.[2] Moreover, only slight variances exist in the costs even though the same cases were selected, since the target values are computed on the basis of similarity scores of the selected cases, not on the cases only.

Rule 3:
IF *resources' condition of target case is plenty*
AND resources' condition of selected case is plenty
THEN *the value of adjusting factor = 1*

Rule 4:
IF *resources' condition of target case is plenty*
AND resources' condition of selected case is medium
THEN *the value of adjusting factor = 1.1*

Rule 5:
IF *resources' condition of target case is plenty*
AND resources' condition of selected case is shortage
THEN *the value of adjusting factor = 1.2*
...

Fig. 5. Example rules for determining the adjusting factor.

Table 5
Test results of the CBR-CURE system

	Test project 1				*Test project* 2				*Test project* 3			
	Scenario (*Minimum similarity score for retrieval*)											
Properties	1 (50)	2 (50)	3 (40)	4 (30)	1 (50)	2 (50)	3 (40)	4 (30)	1 (50)	2 (50)	3 (40)	4 (30)
Site area (m^2)	2020	2020	2020	2020	4040	4040	4040	4040	810	810	810	810
No. of floors above ground	6	6	6	6	5	5	5	5	4	4	4	4
No. of floors below ground	1	1	1	1	2	2	2	2	1	1	1	1
Average floor height (m)	3.3	3.3	3.3	—	3.3	3.3	3.3	—	3.3	3.3	3.3	—
Percentage of window (%)	15	15	15	—	15	15	15	—	15	15	15	—
Percentage of landscaping (%)	20	20	—	—	15	15	—	—	20	20	—	—
Percentage of pavement (%)	30	30	—	—	30	30	—	—	30	30	—	—
No. of elevators	5	—	—	—	6	—	—	—	3	—	—	—
Slope of roof (in degrees)	20	—	—	—	0	—	—	—	0	—	—	—
Labor condition	Plenty	Plenty	Plenty	Plenty	Medium	Medium	Medium	Medium	Shortage	Shortage	Shortage	Shortage
Selected case ID (similarity score)	5(65) 53(57) 26(56)	5(65) 26(56)	5(52) 26(41)	5(46) 26(34)	32(57) 10(56)	32(54) 10(54)	32(48) 10(46)	32(41) 10(39)	2(63) 25(59) 52(54)	2(61) 25(56)	2(46) 25(41)	2(39) 25(34)
Duration difference (%)	7.6	6.3	6.5	6.6	4.2	4.2	4.1	4.1	−6.9	−5.3	−5.4	−5.5
Equipment cost difference (%)	11.3	9.5	9.7	9.9	11.2	11.3	11.2	11.2	−11.6	−9.5	−9.6	−9.7
Machine cost difference (%)	9.6	8.1	8.4	8.6	7.7	7.7	7.6	7.6	−17.6	−15.5	−15.7	−15.8
Labor cost difference (%)	11.1	9.5	9.8	10.0	9.4	9.4	9.3	9.3	−13.3	−11.1	−11.3	−11.4

Note: 1. TCIS's output of test project 1: duration = 285; equipment cost = 233,177; machine cost = 2,350,794; labor cost = 2,006,048.
2. TCIS's output of test project 2: duration = 280; equipment cost = 248,914; machine cost = 2,334,079; labor cost = 2,021,369.
3. TCIS's output of test project 3: duration = 230; equipment cost = 165,865; machine cost = 1,592,441; labor cost = 1,395,638.
4. Difference (%) = $\frac{\text{TCIS(duration, cost)} - \text{generated(duration, cost)}}{\text{TCIS(duration, cost)}} \times 100\%$

5 CONCLUSION

In the experience-oriented construction industry, CBR adapts well for computer applications, as long as information from previous construction projects is available. A major issue in such an application is that proper features to represent a construction project must be considered carefully before implementing the application. If they are not defined properly, the retrieved cases may not lead to a correct solution.

CBR-CURE is a pilot system in applying the CBR approach to the construction planning domain based on 60 hypothetical cases. Even though CBR-CURE is experimental and may not satisfactorily predict real-life construction costs and duration, test results in this study confirm that CBR provides an acceptable alternative for rapid construction planning. Such a merit is primarily owing to the experience-driven nature of construction industry and CBR's ability to mimic the decision-making processes of human planners. Thus CBR provides an alternative in solving experience-oriented problems when traditional ESs or NNs encounter difficulties.

The results presented in this study also provide new research directions for applying CBR to other construction management problems, such as risk analysis, schedule generation, site layout, construction method selection, resource management, and time/cost predictions. The drawback of CBR in blindly using retrieved cases can be compensated for by incorporating similar technologies such as ESs or NNs. The means of incorporating these technologies and integrating CBR with other technologies (such as multiple-criteria decision making and rule-based reasoning) to enhance its problem-solving ability requires further study.

REFERENCES

1. Aamodt, A. & Plaza, E., Case-based reasoning: Foundational issues, methodological variations, and system approaches, *AI Communication*, **7**(1) (1994), 39–59.
2. Björnsson, H. & Hjarøar, B. P., Construction expertise transfer with knowledge based systems. In *Proceedings of First Congress on Computing in Civil Engineering*, *Washington*, *DC*, ASCE, New York, 1994, pp. 964–71.
3. Esteem Software, Inc., *ESTEEM Documentation*, Version 1.4. Cambridge City, IN, 1995.
4. Flemming, U. & Woodbury, R., Software environment to support early phases in building design (SEED): Overview, *Journal of Architectural Engineering*, *ASCE*, **1**(4) (1995), 147–52.
5. Gupta, U. G., How case-based reasoning solves new problems, *Interfaces*, **24**(6) (1994), 110–19.
6. IntelliCorp, Inc., *KAPPA-PC Software*, Version 2.3. Mountain View, CA, 1994.
7. Kolodner, J. L., Improving human decision making through case-based decision aiding, *AI Magazine*, **12**(2) (1991), 52–68.
8. Maher, M. L. & Zhang, D. M., CADSYN: Using case and decomposition knowledge for design synthesis. In *Proceedings of the First International Conference on Artificial Intelligence in Design*, Edinburgh, UK, Butterworth-Heineman, Oxford, 1991, pp. 137–50.
9. Pearce, M., Goel, A., Kolodner, J. L., Zimring, C., Sentosa, L. & Billington, R., Case-based design support: A case study in architectural design, *IEEE Expert*, **7**(5) (1992), 14–20.
10. R.S. Means Company, Inc., *Building Construction Cost Data* 1993, 51st ed., Kingston, MA, 1994.
11. Wang, J., Integrated Case-Based Reasoning for Structure Design, Ph.D. thesis, Department of Civil Engineering, Stanford University, CA, 1991.
12. Watson, I. & Abdullah, S., Developing case-based reasoning systems: A case study in diagnosing building defects, *IEE Digest No*: 1994/057, *Case Based Reasoning*: *Prospects for Applications* (1994), 1/1–1/3.
13. Yang, J. B. & Yau, N. J., Applications of case-based reasoning in construction engineering and management. In *Proceedings of the Third Congress on Computing in Civil Engineering*, *Anaheim*, *CA*, ASCE, New York, 1996, pp. 663–9.
14. Yau, N. J., An Object-Oriented Project Model for Integrating Building Design, Construction Scheduling, and Cost Estimating for Mid-rise Construction, Ph.D. thesis, Department of Civil Engineering, University of Illinois at Urbana-Champaign, IL, 1992.
15. Yau, N. J. & Yang, J. B., A case-based reasoning approach for construction planning. In *Proceedings of the* 13*th International Symposium on Automation and Robotics in Construction*, ISARC, Tokyo, Japan, 1996, pp. 467–74.