

A hierarchical case representation using context guided retrieval

Ian Watson^{a,*}, Srinath Perera^b

^a*AI-CBR, University of Salford, Bridgewater Building, Salford, M5 4WT, UK*

^b*Department of Building Economics, University of Moratuwa, Moratuwa, Sri Lanka*

Received 16 July 1998; accepted 24 July 1998

Abstract

This paper presents a hierarchical case representation using a context guided retrieval method. The performance of this method is compared to that of a simpler flat file representation using standard nearest neighbour retrieval. The estimation of construction costs of light industrial buildings is used as the test domain. Each case in the system has approximately 400 features. These are structured into a hierarchical case representation that holds more general contextual features at its top and specific building elements at its leaves. Problems are decomposed into sub-problems and solutions recomposed into a final solution. Comparative results show that the context guided retrieval method using the hierarchical case representation is significantly more accurate than the simpler flat file representation using standard nearest neighbour retrieval. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Case-based reasoning; Case-representation; Context guided retrieval

1. Introduction

This paper presents results that compare the accuracy of a hierarchical case representation using context guided retrieval with the accuracy of a flat case representation using standard nearest neighbour retrieval. The data presented in this paper is more extensive than that presented in a recent paper by the same authors [1].

Representing cases as a set of constituent pieces [2, 3], snippets [4–6] or footprints [7, 8], instead of as a single large entity, has long been proposed as a way of improving the effectiveness of a CBR system. These parts, when represented as separate structured cases, can be represented, retrieved and recomposed separately to create new solutions [9–12]. Some systems, for example, CADSYN, explicitly take into account the context of a snippet or sub-problem to reduced constraint problems when recomposing solutions [13].

Many successful CBR systems use relatively simple case representations of attribute–value pairs stored in flat files or record structures similar to those of a conventional database. There are good reasons for this. A primary one is, that for many commercial applications, the knowledge engineering effort required to create case-bases must be kept to a minimum. These case representations may be characterised as

being knowledge-poor. That is they do not contain many (or any) structures that describe the relationships or constraints between case features. However, these case representations usually describe relatively simple cases with few indexed features, perhaps in the order of ten to twenty indexed features.

As the number of indexed case features increases (i.e. the number of features that are predictive of a case's solution or outcome) the utility of this knowledge-poor approach reduces. As the problem space increases, from say a 20 dimensional space to a 200 dimensional space it becomes statistically less likely that a close matching case will exist. Thus, a retrieve and propose CBR system (i.e. one without adaptation) may be proposing a relatively distant solution. If adaptation is used, the adaptation effort or distance will increase correspondingly, possibly reducing the accuracy or utility of the solution. This is illustrated in the two figures below, after [14] (page 8). Fig. 1, shows, on the left, a relatively small problem space and assumes a similar sized solution space. Notice that the retrieval distance (the arrow labelled R) and the adaptation distance (the arrow labelled A) are both quite short. As the size of the problem space increases (shown on the right) the retrieval and adaptation distances may increase, as shown by the lengths of the arrows.

Moreover, as has been reported by [15] there is often an inverse relationship between the number of cases in a case-base and the number of indexed features in the cases. This is

* Corresponding author. Tel.: + 44 (0)161-295-5227; Fax: + 44 (0)161-295-5011; e-mail: ian@ai-cbr.org

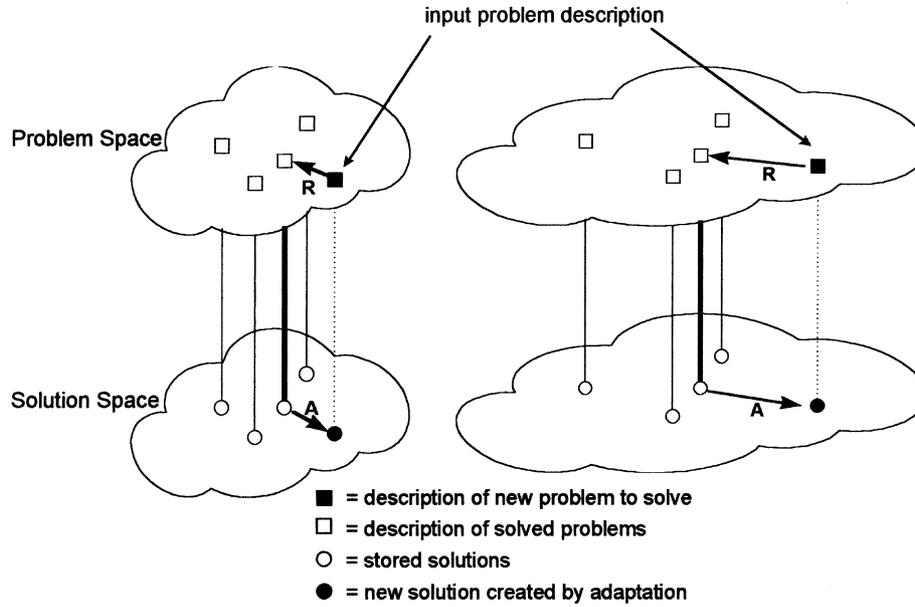


Fig. 1. Small and large problem & solution spaces, after [14] (page 8).

because it is often harder to collect a few large cases than it is to collect hundreds of small cases. Thus, case coverage is often likely to be lower in a large problem space than in a small problem space. This may cause the case-base to return a mediocre match that will require considerable adaptation, resulting in poorer solutions.

A potential solution to this problem is the divide and conquer approach. This suggests that, where suitable, a large problem is divided into several smaller sub-problems,

each of which can be solved separately using CBR. The sub-solutions can then be combined to produce an accurate solution to the entire problem [13]. A key assumption for this approach is that the sub-problems are not highly constrained one upon the other, so that they can be solved independently (i.e. that the problem can be sensibly decomposed and the solution recomposed). This approach may be visualised as in Fig. 2.

The advantage of this approach is that each individual

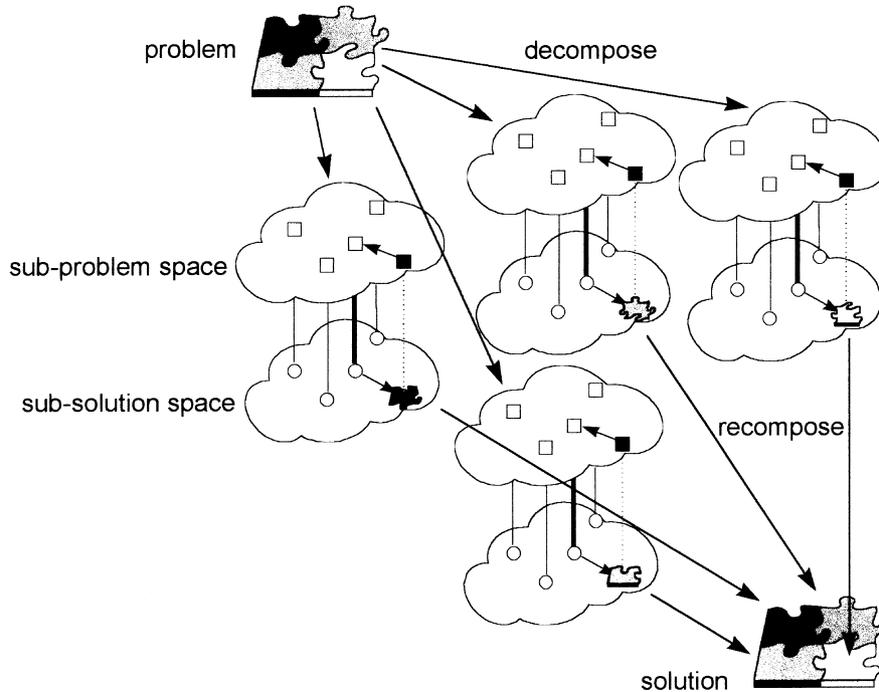


Fig. 2. Problem decomposition and solution recombination.

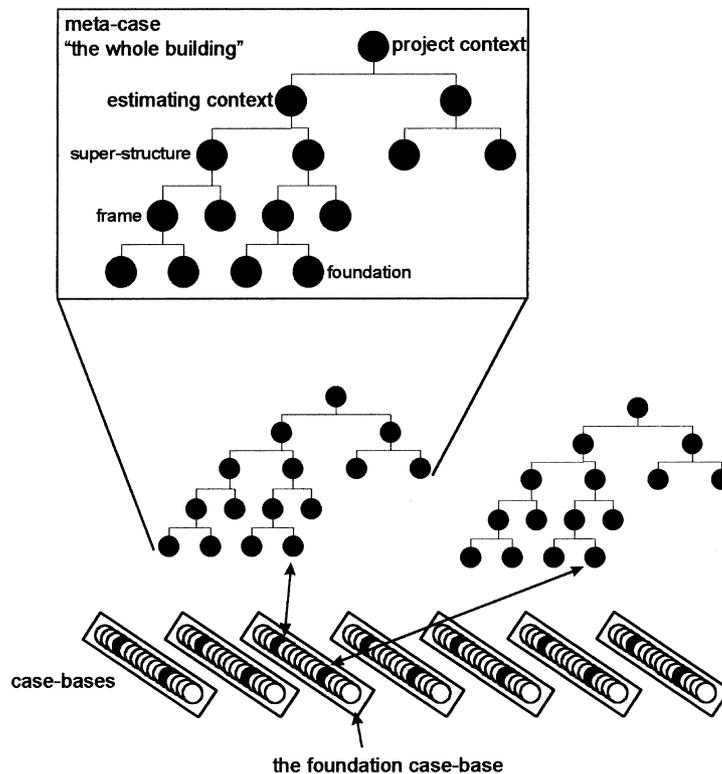


Fig. 3. Schematic of the hierarchical case representation.

sub-problem is represented by a case-base that is significantly smaller (in terms of problem and solution space size) than if the whole problem were represented by a single case-base. Because each sub-problem space has fewer case features, the theory predicts, that each individual sub-case retrieval distance will be shorter than for the un-decomposed problem. Therefore, the adaptation distance will be shorter and a better sub-solution will be generated. Assuming there are no conflicting constraints, the recomposition of sub-solutions will produce a better solution than would have been obtained by using a single large case-base. One way that has been suggested to reduce constraint problems with solution recomposition is to use contextual information to guide retrieval [15–20]. The argument being, that if cases share similar contexts, this will reduce constraint problems during solution recomposition.

The purpose of the study presented in this paper was to quantitatively assess the accuracy of a CBR system that uses a hierarchical case representation and context-guided retrieval to decompose a complex problem and recompose a solution. The accuracy of this complex case representation and retrieval technique is compared to that of a simple flat record of attribute–value pairs using a standard nearest neighbour retrieval algorithm. The evaluation will show that the more complex representation and retrieval method outperforms the simpler representation, thereby justifying the knowledge engineering and programming effort put into it.

2. The problem domain

For this study we selected the estimation of the construction costs of light industrial warehousing as a suitable domain. These buildings are used as storage and distribution warehouses, as low cost retail buildings, and as light industrial factory units. They were suitable for this study for the following reasons:

- Warehouses are strictly functional buildings with aesthetic issues being very secondary (i.e. they rarely win design prizes). Consequently, cost is a more important issue than for most other building types.
- They are constructed using steel frames that are produced in standard sizes along with many other components (e.g. roofing sheets) that are also produced in standard sizes.
- The buildings are structurally fairly simple and consequently the constraints between different building elements are small. This therefore suggested that divided and conquer would be appropriate.
- The cost of a building is derived directly from the cost of its sub-assemblies. Thus, the problem decomposes naturally. This is supported by the way that cost estimators usually work. They calculate the cost of each sub-assembly and sum them to obtain a total cost.
- Finally, we had access to a cost estimating computer system for this building type. This has significant methodological importance and will be discussed later.

Table 1
A selection of attributes from the project context case definition

| | Attribute | Value(s) | Data Type |
|-----|---------------------|--|-----------------------|
| 1. | Case_No | Value per project | cat-nir:capitols |
| 2. | Number_key | Unique integer value per case per case-base | cat:integer-or-nil |
| 3. | Source_cases | List of cases | default |
| 4. | Name_of_Project | Text | default |
| 5. | Site_Address | Text | default |
| 6. | Site_Post_Code | Text | default |
| 7. | Client | Text | default |
| 8. | Client_Address | Text | default |
| 9. | Client_Post_Code | Text | default |
| 10. | Type_of_warehouse | StorageDistributionRetail | catnl:wh-type |
| 11. | Type_of_occupier | Owner occupierTenant occupierDeveloper | catnl:occupier |
| 12. | Use | Basic materialsCompleted — goodsPerishable — goodsEquipmentRetail goodsElectronic — ComponentsRe-distribution — Goods | catnl:use |
| 13. | Storage_Category | FlammableNon-flammableChemical FlammableChemical Non-flammable | catnl:st-cat |
| 14. | Region | List of regions (BCIS) | catnl:regions |
| 15. | Construction_period | Months | cat:float-or-nil |
| 16. | Project_duration | Months | cat:float-or-nil |
| 17. | Project_Cost_limit | £ | cat:integer-or-nil |
| 18. | Tender_Month | List of Months | catnl:months |
| 19. | Tender_Year | 1983–1998 | cat-nir:year1983-1998 |
| 20. | Actual_project_cost | £ | cat:integer-or-nil |
| 21. | Total_variations | £ | cat:integer-or-nil |
| 22. | Completed_duration | Months | cat:float-or-nil |
| 23. | Completed_date | Text | default |
| 24. | Type_of_contract | List of Contract Types | catnl:contract |
| 25. | Gross_Floor_Area | m ² | cat:integer-or-nil |
| 26. | Gross_office_area | m ² | cat:integer-or-nil |
| 27. | Type_of_Structure | Portal FramePropped Portal — FrameSteel Frame and JoistsClear Span — FrameStructural Steel — FrameTimber Frame | catnl:struct-type |
| — | — | — | — |
| 66. | Structural_Engineer | Text | default |
| 67. | Services_Engineer | Text | default |
| 68. | Other_Consultants | Text | default |
| 69. | Contractor | Text | default |

3. The case representation

The system, called NIRMANI, was implemented in ART*Enterprise, from BrightWare (<http://www.brightware.com>), on Windows 95 [21]. The environment provides an object-oriented knowledge-based development environment, that supports objects, rules (a forward chaining Rete algorithm), a procedural programming environment, case-based reasoning (nearest neighbour), a GUI builder, and an ODBC database interface [22]. Representing cases hierarchically is a popular approach to the use and reuse of sub-cases (e.g. [5, 23, 24]). A building in NIRMANI is a meta-case, consisting of a hierarchy of cases and sub-cases. At the top of the hierarchy is the *Project Context* case. The second level contains the *Architectural Context* and *Estimating Context* cases representing the perspectives (or views) of architects and cost estimators. A third level decomposes the design into functional spaces and aesthetic requirements hierarchies and the estimating problem into an industry standard elemental classification hierarchy [25].

Each node in the hierarchy is stored in a separate case-base (shown schematically in Fig. 3). The cases are stored as records in a relational database external to the system since this has the benefit of allowing a design organisation to keep their case data in their existing databases [26]. An object hierarchy within the system maps to the tables in the database and cases are presented (when required) as instances. Cases contain attributevalue pairs as case features, as in Table 1.

A *Project Context* case describes the environment within which the project was earned out (features such as the type of building, its intended function, gross internal floor area (GIFA), the site conditions, and other features common to the project context). The second level cases (architectural and estimating) describe the context of the sub-problems. The system prefers to retrieve sub-cases with similar contexts (i.e. with similar parents in the hierarchy) in order to reduce problems of case adaptation and solution recomposition due to contextual dissimilarity. The interface of NIRMANI allows cases to be viewed as attribute-value

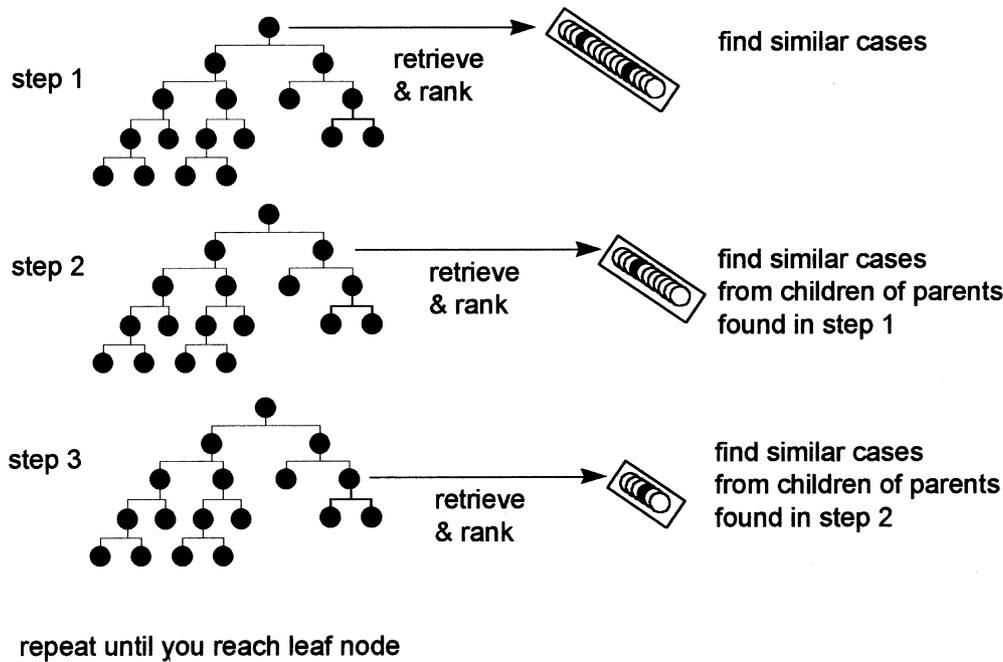


Fig. 4. Context guided retrieval.

pairs along with CAD drawings and other multimedia elements. It supports case comparison using a tabulated form similar to a spreadsheet.

4. Retrieval

NIRMANI provides a variety of retrieval methods, of which only two are compared in this paper. Full details of these retrieval methods can be found in [25]. ART*Enterprise uses a nearest neighbour algorithm with weighted features. Its programming environment gives the developer considerable control of the algorithm making it a good environment to explore different retrieval strategies. The two strategies compared in this paper are described below.

4.1. Default retrieval

This is essentially standard nearest neighbour retrieval. The user is allowed to select which features are indexed. These will usually be the majority of the features in the *Project Context* case (except the construction cost) plus some other significant features from other aspects of the building. For example, the user may want a glazed curtain wall on the front elevation of the building but have no definite views or wishes as to the roofing type. The user may set weights on features reflecting their relative importance to them.

In default retrieval an index is prepared dynamically at run-time for those case features entered by the user. Feature comparison is carried out as in normal nearest neighbour retrieval. A normalised match score for each entire meta-case is calculated and the highest ranking cases are then

presented to the user. Only an entire meta-case can then be selected for adaptation.

4.2. Context guided retrieval

Context guided retrieval proceeds in series of recursive steps down the hierarchy of the case representation. In the first step, the features of the *Project Context* case (at the top of the hierarchy) are used to retrieve similar *Project Context* cases from the *Project Context* case-base. This is done using ARM's standard nearest neighbour algorithm. In the second step, retrieval of cases from the *estimating* or *architectural* case-bases (the next nodes down the hierarchy) is restricted to those cases that are the children of the cases found similar in the first retrieval step. That is, retrieval is limited to those sub-cases that share similar project contexts (i.e. similar parents). This process is repeated all the way down the hierarchy. Retrieval at each level is restricted to those cases in a case-base that have similar parents (shown schematically in Fig. 4).

This process reduces the search space by enforcing contextual similarity. However, if a close enough match cannot be found at any level (this is more likely to occur at leaf nodes since the number of cases included in the search may reduce significantly at each level) then the contextual guiding can be relaxed. This relaxation is achieved by back tracking up the hierarchy and reducing the threshold at which similarity is judged acceptable for the parent case. This will increase the number of cases allowed into the children's retrieval process. This relaxation can proceed all the way to zero, if necessary, allowing

retrieval from all cases in a child's case-base, thus removing the context guidance completely.

5. Adaptation

Cases are ranked and presented to the user. Users are allowed to select cases and case features for adaptation. Note that using the default retrieval method only sub-cases from one meta-case can be used for adaptation. Whereas, for context guided retrieval, sub-cases from different meta-cases with a similar context can be used. Moreover, using context guided retrieval adaptation can occur at the elemental unit level of detail, whereas for the default retrieval adaptation occurs at the level of the project context case (i.e. only the total estimated construction cost is adapted). A modification knowledge-base, containing a set of rules, functions and procedures provides the adaptation. In general, adaptation is in the form of parameter adjustment through interpolation. For example, if a retrieved case has the feature 'floor finishes' at a cost of '£12 000' with a GIFA of '2000 m²', then the adaptation function will calculate a rate for floor finishes of '£6 per m²'. This rate can then be applied to a new case with a different GIFA but a similar specification for floor finishes.

6. Methodology

In the 1980s and early 1990s Salford University, in collaboration with the Royal Institution of Chartered Surveyors (the RICS is the professional institution for cost estimators in the UK), developed several knowledge-based construction cost estimation systems. The first of these, a rule-based system called ELSIE, could estimate the construction costs of commercial office developments [27]. In a subsequent development another rule-based system, called ELI, was developed for estimating the construction costs of light industrial warehouse units. These systems are sold commercially, by a joint venture company, and have sold over a thousand copies world-wide.

The RICS commissioned a study to check the accuracy of the systems [28], which found that their estimates are within plus or minus 5% of eventual construction costs. This is well within acceptable error and is as good as the most experienced cost estimators [29]. For our study we used ELI as both a case generator (i.e. to produce projects to populate our case-base) and as an evaluator (i.e. to test the accuracy of the CBR systems).

6.1. Case acquisition

Details of sixty construction projects were obtained from the Building Construction Cost Information Service (BCIS), an information service for the UK construction industry. ELI was used to generate a further twenty hypothetical construction projects. These projects were carefully

designed to fill in the gaps between the sixty real projects from the BCIS. These were entered into a database that NIRMANI used for its case data. The projects generated by ELI were carefully designed so as to create a case-base with an even case distribution. Thus, projects were created which had a variety of functions (e.g. dry goods distribution warehouses, cold storage warehouses, flammable goods storage and distribution, retail warehouses, etc.). The projects varied in size consistently in graduations of approximately 100 m², from 1,500 m² to 5,000 m². In addition, a range of construction complexity with additional features, such as office space, were included. We recognise that this case-base is artificial. We felt that a well distributed case-base should be analysed before attempting a randomly distributed one.

6.2. Evaluation

Evaluation of the accuracy of NIRMANI using the two retrieval techniques described above was performed as follows:

New projects (i.e. ones that NIRMANI had never seen) were developed by ELI and hence we knew ELI's estimation of their construction cost. These were then presented to NIRMANI as new problems for it to estimate. New projects were developed in increments of 250 m² from 1500 m² to 5000 m². Five projects were developed in each size range, each with a different function and building complexity (e.g. material stored, office space, loading bays, etc.)

The results from the evaluation tests were statistically analysed using the coefficient of variation method. This technique is widely used as the most common criteria for the determination of the accuracy of an estimating method or model [29]. CV is defined as:

$$CV = \frac{\text{Standard Deviation of Residuals } (S_r)}{\text{Mean Cost of All Schemes} - \text{Actual } (M_a)}$$

Thus, CV can be termed as the estimating error where: accuracy = 100 - % estimating error, and therefore: accuracy = 100 - CV.

7. Results

A summary of the results is shown in Fig. 5. The graph shows the average CV of each of the two techniques. Exactly the same feature weightings were used for both the NN retrieval and the context guided retrieval. An average CV of zero would indicate perfect accuracy.

A major study of the accuracy of estimation in the construction industry revealed that an accuracy ranging from ± 15% to ± 20% is acceptable for early stage estimating of construction costs [29]. Therefore, the majority of the estimates using context guided retrieval were well within acceptable error, whilst only a few approach the ± 20% limit. However, the flat representation using standard nearest neighbour failed in the majority of tests. The accu-

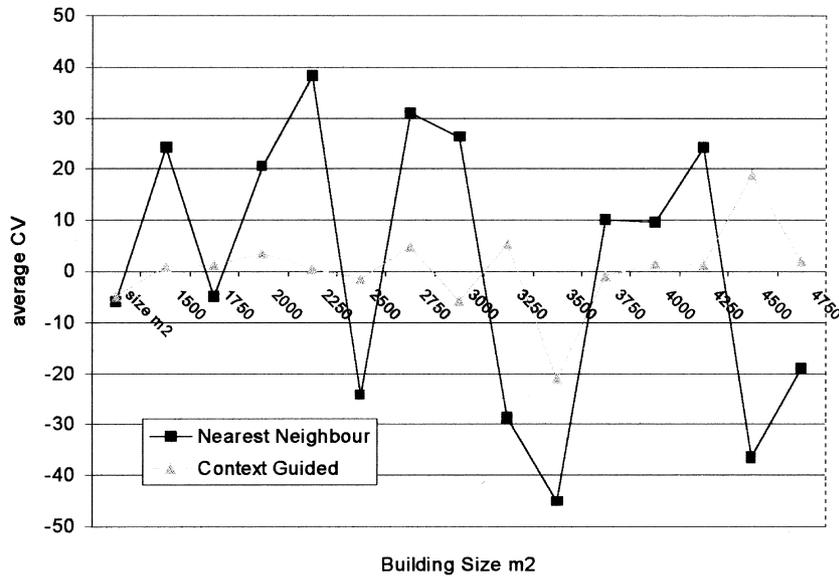


Fig. 5. The average CV of nearest neighbour and context guided retrieval.

racy of the context guided retrieval is increased because it can find nearest neighbours for individual elements of buildings, whereas the other technique cannot find a whole building that matches well enough.

Because the sample size was low (i.e. $n = 85$), the Students' t -test was used for statistical analysis. The aim of this test is to determine whether the results obtained for the standard nearest neighbour and context guided retrieval represent significantly different approaches. In statistical terms this involves testing whether the test samples could be from the same population. In order to achieve these results a 'Paired Sample Student's t -Test' was carried out. The test hypothesis was as follows: $H_0: \mu = 0$ (the mean of the difference between the two techniques is zero). t -Tests were carried out for a 95% level of confidence, which is accepted as indicating statistical significance. This found that H_0 could be rejected at 95% confidence levels.

8. Conclusion

The systematic evaluation of a CBR system is very difficult because such systems are typically very complex with many interacting components [30]. Consequently, this study has simplified the performance of our system down to a single quantifiable measure -estimating accuracy. We accept that this measure is a simplification of the performance of our system. Nonetheless, the evaluation demonstrates that the context guided retrieval method outperforms that of the simpler flat-file nearest neighbour method. The only times that the simpler technique performed acceptably were when a problem happened to find a close near neighbour within the case-base. When the simpler technique performed badly it was because it was unable to find a complete matching case and was forced to use the closest

case that matched on a subset of features. Conversely, when the context guided retrieval method significantly outperforms the simpler technique it is because it has composed a solution from many cases. Thus, when a close near neighbour cannot be found the divide and conquer approach, using context guided retrieval, performs better as the theory predicts. It is interesting to note that the simpler technique usually recognises which case can contribute most to solution, but, by being unable to use snippets from other cases as well, its accuracy is reduced.

We recognise that this is still a fairly limited study, although the sample size is much greater than that reported in [1]. We have shown that for our tests the context guided retrieval is more accurate than the standard nearest neighbour retrieval. Moreover, there is a 95% confidence that this technique is statistically different from the standard nearest neighbour retrieval. The number of tests is still limited and therefore it would be unwise to rely too heavily on the simple statistical analysis performed here. However, the results are indicative and support the view that divide and conquer, through problem decomposition and solution recomposition, is an effective method of solving problems with large complex cases. The context guided retrieval method evaluated here may also be a useful way of reducing the problems of conflicting constraints between parts of the solution. The fact that the case-base was populated with an evenly distributed set of cases may have skewed our results. Although from the results it would appear that this should skew the results in favour of the simpler method. Since it performs better when a close good match can be found, one would expect it to perform more erratically with a more unevenly distributed case-base. Finally, it was interesting to see that the case-based estimator performed as well as the rule-based estimation system, with a mean error of 2%. The rule-based estimator took over three person years to

implement, whilst the case-based estimator took less than half that time. This further supports the many findings that show that CBR systems can be implemented quicker than their rule-based counterparts [31, 32].

Acknowledgements

The authors would like to acknowledge the Association of Commonwealth Universities, the University of Moratuwa, Sri Lanka, and EPSRC, whose grants: GR/J42496, GR/J43660 and GR/L16330, helped support this work.

References

- [1] I. Watson, R.S. Perera, The evaluation of a hierarchical case representation using context guided retrieval, in: *Case-Based Reasoning Research and Development, The proceedings of the 2nd International Conference on Case-Based Reasoning, Lecture Notes In Artificial Intelligence*, Springer, New York, 1997.
- [2] R. Barletta, W. Mark, Breaking cases into pieces, in: J.L. Kolodner (Ed.), *Proceedings of the DARPA Case-Based Reasoning Workshop*, Morgan Kaufmann, California, US, 1988.
- [3] L. Macedo, F.C. Pereira, C. Grilo, A. Cardoso, Plans as structured networks of hierarchically and temporally related case pieces, in: I. Smith, B. Faltings (Eds.), *Advances in Case-Based Reasoning, Lecture Notes in Artificial Intelligence 1168*, Springer, Berlin, 1996.
- [4] J.L. Kolodner, Retrieving events from a case memory: a parallel implementation, in: J.L. Kolodner (Ed.), *Proceedings of the DARPA Case-Based Reasoning Workshop*, Morgan Kaufmann, California, US, 1988.
- [5] M. Redmond, Distributed cases for case-based reasoning: facilitating use of multiple cases, in: *Proceedings of AAAI, AAAI/MIT Press*, 1990.
- [6] K.P. Sycara, D. Navinchandra D, Influences: a thematic abstraction for creative use of multiple cases, in: E.R. Bareiss, *Proceedings of the DARPA Case-Based Reasoning Workshop*, Morgan Kaufmann, California, US, 1991.
- [7] M.M. Veloso, *Learning by Analogical Reasoning in General Problem Solving*, Carnegie Mellon University, School of Computer Science Technical Report no. CMU-CS-92-174, 1992.
- [8] C. Bento, L. Macedo, E. Costa, E., RECIDÉ — Reasoning with cases imperfectly described and explained, in: *Proceeding of the 2nd European Workshop on Case-Based Reasoning, EWCBR2*, Chantilly, France, 1994, pp. 3–17.
- [9] U. Flemming, Case-based design in the SEED system, *Automation in Construction 3*, Elsevier Science B.V., 1994, pp. 123–133.
- [10] M.L. Maher, B. Balachandran, B., Flexible retrieval strategies for case-based design, in: J.S. Gero, F. Sudweeks (Eds.), *Artificial Intelligence in Design '94*, Kluwer Academic Publishers, Netherlands, 1994, pp. 163–180.
- [11] B. Bartsch-Sporl, Towards the integration of case-based, schema-based and model-based reasoning for supporting complex design tasks, in: M. Veloso, A. Aamodt (Eds.), *Case-Based Reasoning Research and Development — ICCBR '95*, Springer, Berlin, 1995, pp. 145–156.
- [12] L. Hunt, R. Miles, Towards an Intelligent Architectural Design Aid, *Expert Systems 12 (3) (1995) 209–218*.
- [13] M.L. Maher, D.M. Zhang, CADSYN: using case and decomposition knowledge for design synthesis, in: J.S. Gero (Ed.), *Artificial Intelligence in Design*, Butterworth-Heinemann, Oxford, UK, 1991.
- [14] D.B. Leake, CBR in context: the present and future, in: D.B. Leake (Ed.), *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, AAAI Press/The MIT Press, Menlo Park, CA, US, 1996.
- [15] M.L. Maher, M.B. Balachandran, D.M. Zliang, *Case-Based Reasoning in Design*, Lawrence Erlbaum Associates, US, 1995.
- [16] K.J. Hammond, CHEF: A model of case-based planning, in: *Proc. American Association for Artificial Intelligence, AAAI-86*, Philadelphia, PA, US, 1986.
- [17] D. Hennessy, D. Hinkle, Applying Case-Based Reasoning to Autoclave Loading, *IEEE Expert 7 (5) (1992) 21–26*.
- [18] J.L. Kolodner, *Case-Based Reasoning*, Morgan Kaufmann Publishers, California, US, 1993.
- [19] R. Marir, I. Watson, Representing and indexing building refurbishment cases for multiple retrieval of adaptable pieces of cases, in: M. Veloso, A. Aamodt, (Eds.), *Case-Based Reasoning Research and Development — ICCBR '95*, Springer, Berlin, 1995, pp. 55–66.
- [20] A. Ram, A.G. Francis, Multi-plan retrieval and adaptation in an experience-based agent, in: D.B. Leake (Ed.), *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, AAAI Press/The MIT Press, Menlo Park, CA, US, 1996.
- [21] I. Watson, R.S. Perera, NIRMANI: a case-based expert system for integrated design and estimating, in: A. Macintosh, C. Cooper (Eds.), *Applications and Innovations in Expert Systems III*, SGES Publications, UK, 1995, pp. 335–348.
- [22] I. Watson, *Applying Case-Based Reasoning: Techniques for Enterprise Systems*, Morgan Kaufmann, CA, USA, 1997.
- [23] A. Goel, K. All, M. Donnellan, A. Garza, T. Callantine, Multistrategy Adaptive Navigational Path Planning, *IEEE Expert 9 (6) (1994) 57–65*.
- [24] D. Alia, K. Branting, Stratified case-based reasoning: reusing hierarchical problem solving episodes, in: *Proceedings of the 13th International Joint Conference on Artificial Intelligence, IJCAI-95*, IJCAI, Menlo Park, CA, US, 1995, pp. 384–390.
- [25] R.S. Perera, I.D. Watson, Multi-agent collaborative case-based estimating and design in NIRMANI: organising a multi-perspective case memory, in: B. Kumar (Ed.), *Information Processing in Civil and Structural Engineering Design*, CivilComp Press, Scotland, 1996, pp. 53–64.
- [26] M. Brown, I. Watson, N. Filer, Separating the cases from the data: towards more flexible case-based reasoning, in: M. Veloso, A. Aamodt (Eds.), *Case-Based Reasoning Research and Development, Lecture Notes in Artificial Intelligence 1010*, Springer, Berlin, 1995.
- [27] P. Brandon, A. Basden, I. Hamilton, J. Stockley, *The Strategic Planning of Construction Projects*, RICS Publications, London, UK, 1988.
- [28] A.M. Castell, A. Basden, G. Erdos, P. Barrows, P.S. Brandon, Knowledge based systems in use: a case study, in: *Proceedings of the 12th Annual Conference of the British Computer Society Specialist Group on Expert Systems*, Cambridge University Press, Cambridge, UK, 1992.
- [29] A. Ashworth, R.M. Skitmore, Accuracy in Estimating, *The Chartered Institute Of Building, Occasional paper no. 27*, 1982.
- [30] J.C. Santamaria, A. Ram, Systematic evaluation of design decisions in case-based reasoning systems, in: D.B. Leake (Ed.), *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, AAAI Press/The MIT Press, Menlo Park, CA, US, 1996.
- [31] E. Simoudis, J.S. Miller, The application of CBR to help desk applications, in: R. Bareiss (Ed.), *Proceedings of the DARPA Workshop on Case-Based Reasoning*, Morgan Kaufmann, San Francisco, CA, US, 1991.
- [32] W. Mark, E. Simoudis, D. Hinkle, Case-based reasoning: expectations and results, in: D.B. Leake (Ed.), *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, AAAI Press/The MIT Press, Menlo Park, CA, US, 1996.