

Automatically Selecting Strategies for Multi-Case-Base Reasoning*

David B. Leake¹ and Raja Sooriamurthi²

¹ Computer Science Department, Indiana University, Lindley Hall 215
150 S. Woodlawn Avenue, Bloomington, IN 47405, U.S.A.
leake@cs.indiana.edu

² Kelley School of Business, Indiana University, BU540
1309 East 10th Street, Bloomington, IN 47405, U.S.A.
raja@indiana.edu

Abstract. Case-based reasoning (CBR) systems solve new problems by retrieving stored prior cases, and adapting their solutions to fit new circumstances. Traditionally, CBR systems draw their cases from a single local case-base tailored to their task. However, when a system's own set of cases is limited, it may be beneficial to supplement the local case-base with cases drawn from external case-bases for related tasks. Effective use of external case-bases requires strategies for *multi-case-base reasoning* (MCBR): (1) for deciding when to *dispatch* problems to an external case-base, and (2) for performing *cross-case-base adaptation* to compensate for differences in the tasks and environments that each case-base reflects. This paper presents methods for automatically tuning MCBR systems by selecting effective dispatching criteria and cross-case-base adaptation strategies. The methods require no advance knowledge of the task and domain: they perform tests on an initial set of problems and use the results to select strategies reflecting the characteristics of the local and external case-bases. We present experimental illustrations of the performance of the tuning methods for a numerical prediction task, and demonstrate that a small sample set can be sufficient to make high-quality choices of dispatching and cross-case-base adaptation strategies.

1 Introduction

Case-based reasoning systems solve new problems by retrieving cases from a case-base of prior experiences, and adapting their solutions to fit new circumstances. As a CBR system initially builds up experiences, its case library may be small, potentially limiting its performance. If external case-bases already exist for similar tasks, drawing on those case-bases may help overcome the system's initial knowledge gaps. Even in a system with an extensive case library, external case-bases may contain specialized expertise that would be useful when handling problems outside of the system's normal range of tasks. Unfortunately, it may be difficult to know when to draw on an external case-base and which external case-bases to access. It may also require additional effort to apply cases from an external case-base, due to inter-case-base differences reflecting differences in tasks or execution environments. For example, if an e-commerce application

* This research is supported in part by NASA under award No NCC 2-1216.

quotes prices based on the product case-bases of different stores, one of which lists prices in dollars and the other in euros, adjustments will be needed to convert prices from the American case-base for European use. Thus, making effective use of the information in multiple case-bases requires a reasoning process that we call *multi-case-base reasoning*: reasoning about when to access external case-bases and how to apply their cases. The two central requirements are *case dispatching* processes, to determine the case-base to which to send a problem, and *cross-case-base adaptation* processes, to adjust solutions from one case-base in response to differences between the two case-bases. The success of these processes depends on selecting the right strategies for particular case-bases and task domains. This paper develops methods by which an MCBR system can automatically choose between case-bases and select useful cross-case-base adaptation strategies, and tests them for a numerical prediction task.

In principle, developing high-quality case dispatching and cross-case-base adaptation strategies could require arbitrarily-large amounts of knowledge about the task domain and available case-bases. However, MCBR may be useful even with imperfect dispatching strategies and cross-case-base adaptation knowledge. If the local case-base has only sparse coverage of the problem space, and an external case-base has extensive coverage, the benefits of accessing cases for more similar problems can sometimes counterbalance errors introduced by differences in solution characteristics, with a net performance improvement compared to relying on the local case-base alone [5]. The central question examined in this paper is how an MCBR system can predict whether MCBR will be useful and which MCBR strategies will be most effective. The paper presents and evaluates strategies for automatically tuning an MCBR system by determining case dispatching criteria and choosing between alternative cross-case-base adaptation strategies. The self-tuning methods calibrate a multi-case-base reasoning system by testing alternative methods on a small set of cases in the problem-stream and selecting strategies that do well on those tests.

The paper begins by discussing the motivations for multi-case-base reasoning and the factors affecting its performance. It then describes two dispatching strategies, and three simple methods for cross-case-base adaptation of cases with numerical solutions. These cross-case-base adaptation methods are applicable to cases which share a common representation, but whose suggested solutions differ due to differences in their task environments. The paper then presents experiments illustrating some of the factors involved in the performance of MCBR and the benefits of self-calibration. The paper closes with a discussion of the broader ramifications of this type of approach to maintaining case-based reasoning systems.

2 Motivations for Multi-Case-Base Reasoning

The success of CBR systems depends on their access to relevant stored cases. CBR research and applications generally assume that the only case source is the reasoner's own case-base, and focus on developing and refining a single, unified case-base. However, growing numbers of deployed case-based reasoning systems promise new opportunities for CBR systems to supplement their local case-bases, by drawing on the case-bases of other CBR systems addressing related tasks. Just as thousands of topic-specific informa-

tion sources are now available on the Web, multiple case-bases may eventually provide large-scale distributed, sharable information resources for CBR.

One way to exploit multiple case-bases might be to take an “eager” approach, merging and standardizing all their cases in preparation for future use. This is not always possible—the owners of different case collections will not necessarily be willing to contribute. Even when case-bases could be merged, however, MCBR’s “lazy” approach of on-demand access may provide significant advantages [5, 6]. For very large case-bases, or when local storage is inadequate to store all cases (e.g., for a substantial case-base stored in a PDA), MCBR enables storing frequently-used cases locally and storing less-commonly-used cases externally. When different sources generate the different case-bases, keeping the case-bases distinct also facilitates a natural division of maintenance effort, with each source maintaining its own case-bases, and other users automatically benefitting as they retrieve their cases from the latest versions of the external case-bases. If the relationships between local and external case-bases change, adjustments are simple: only cross-case-base adaptation policies must be changed, rather than all the stored cases (e.g., if the dollar-euro exchange rate changes, only the cross-case-base adaptation procedure for prices must be changed).

The primary focus of this paper, however is on a use of MCBR that applies regardless of case-base size and maintenance concerns: using MCBR to guide case acquisition from related external case-bases, when a CBR system is building up its initial case-base. Compared to simply gathering and merging cases from all available case-bases, MCBR has three main advantages for this task. First, MCBR selectively adds only those cases needed to solve the problems the system actually encounters, keeping the case-base more compact. (For example, a New Yorker with a case-based restaurant advisor on a PDA may only need cases for New York; cases for other regions need not be merged unless—and until—the need arises.) Second, avoiding eager merging gives the flexibility to draw on any new case-bases that may become available, possibly enabling the system to choose to use higher-quality cases than would have been imported by eager merging. Third, when cross-case-base adaptation is required, using MCBR to import cases as needed can improve solution quality compared to performing cross-case-base adaptation on all external cases and eagerly merging them. Because cross-case-base adaptation may be imperfect, sometimes solving a problem using a local case for a less-similar problem will give better results than solving it using a cross-case-base-adapted version of an external case generated for a more similar problem. The case-dispatching strategies of MCBR take this into account in their decisions about when to draw on external case-bases, but eager merging—by simply performing cross-case-base adaptation on all cases and merging the case-bases—loses that capability.

3 Factors Affecting Performance of Multi-Case-Base Reasoning

Whether it is beneficial to draw on an external case-base depends primarily on five factors. The usefulness of MCBR is increased if:

1. **The external case-base has needed competence that the local case-base lacks:** If the local case-base lacks cases sufficiently similar to a problem, it can be worthwhile to dispatch the problem to a case-base with similar cases, even if cross-case-

base adaptation is weak—the benefits of more similar prior cases may counterbalance errors introduced by cross-case-base adaptation.

2. **The local and external case-bases have similar representations:** If local and external case-bases use dissimilar representations, conversion from one to another may significantly increase the burden for MCBR. Our methods assume matching representations, to focus on issues in dispatching and cross-case-base adaptation, but we return to this issue in Section 6.
3. **Cases for similar problems in the local and external case-bases have similar solutions:** In this situation, minimal cross-case-base adaptation will be needed.
4. **Cross-case-base adaptation is strong and/or local adaptation is weak:** High-quality local adaptation procedures can provide good solutions despite sparse local coverage, decreasing the need for dispatching. Conversely, high-quality cross-case-base adaptation can effectively augment the local case-base with high-quality external cases, increasing the benefit of dispatching, and helping to overcome problems with weak local adaptation knowledge.
5. **External case-bases are easily accessible:** Accessing an external case-base may impose additional costs reflecting factors such as access fees, access delays, and bandwidth used. These factors must be balanced with quality considerations when deciding whether to dispatch cases to external case-bases.

The experiments in Section 5 will explore how different levels of local competence affect the overall benefits of dispatching, given weak cross-case-base adaptation knowledge, and how learning can assist in automatically selecting dispatching and cross-case-base adaptation strategies. The next section describes our strategies for cross-case-base adaptation, case dispatching, and automatic self-calibration.

4 Strategies for Multi-Case-Base Reasoning

Applying MCBR requires augmenting standard CBR with mechanisms to decide when to dispatch problems to external case-bases, where to dispatch them, and how to perform cross-case-base adaptation of the returned cases.

4.1 Samples of Knowledge-Light Cross-Case-Base Adaptation Strategies: Global and Local Interpolations

To illustrate the self-calibration process, we consider how it can be used to choose between four cross-case-base adaptation strategies for the task of case-based numerical prediction. One of these is the baseline of the identity (i.e., no cross-case-base adaptation); the other three are simple “knowledge-light” methods that can be derived from the case-bases involved, without requiring domain-specific knowledge.

1. **No cross-case-base adaptation (ID):** This uses the identity function for cross-case-base adaptation, giving a baseline for comparison with other methods.
2. **Linear interpolation, based on global min and max (Minmax):** This method pairs the minimum and maximum solution values in both case bases, and estimates

CBA1($P, local_CB, external_CB$)

- 1 \triangleright Calculate solution in external CB
- 2 $external_soln \leftarrow$ solution to P using $external_CB$
- 3 \triangleright Calculate scale factor for cross-case-base adaptation
- 4 $reference_external_case \leftarrow \operatorname{argmin}_{C \in external_CB} dist(P, C)$
- 5 $reference_external_prob \leftarrow$ problem part of $reference_external_case$
- 6 $reference_external_soln \leftarrow$ soln to $reference_external_prob$ using $external_CB$
- 7 $reference_local_soln \leftarrow$ solution to $reference_external_prob$ using $local_CB$
- 8 $scale_factor \leftarrow reference_local_soln / reference_external_soln$
- 9 \triangleright Return external solution scaled for consistency with $local_CB$
- 10 Return $external_soln * scale_factor$

Algorithm 1. CBA1. This method scales external solutions to fit the local case-base, using a scale factor based on comparing the external and local solutions for a similar reference problem already in the external case-base.

conversions of values in the external case-base by linear interpolation to the local case-base. This strategy will be usable if the extreme values of the case-bases are available, and will be effective to the extent that (1) there is a correspondence between extremes in both case-bases, and (2) the correspondence between intermediate values can be approximated as globally linear.

3. **Local approximation, based on calculating a scale factor starting from the most similar external case (CBA1):** This method calculates a new scale factor for each problem. Given an input problem, it retrieves the external case C for the most similar problem. The problem solved by C is then re-solved in the local case-base, and a scale factor is determined by the ratio of the local and external solutions. The solution of the input problem is calculated by solving it in the external case-base, and multiplying the result by the scale factor (Algorithm 1).
4. **Local approximation, based on calculating a scale factor starting from the most similar local case (CBA2):** This method is analogous to CBA1, but calculates the scale factor based on the most similar local case.

4.2 Case Dispatching and Calibration

We consider two case dispatching strategies. The first, *threshold-based dispatching* [5], dispatches problems to an external case-base if local cases are not available for sufficiently similar problems. This method can be used to decide whether a problem is processed locally or dispatched to an external case-base. Because performance depends on setting the right dispatching threshold for the case-bases and cross-case-base adaptation strategies, we also describe an algorithm for setting the dispatching threshold based on a set of test problems. The second strategy, *case-based dispatching*, dispatches problems to the case-bases that best solved similar problems in the past. This strategy can be used to dispatch to an arbitrary number of case-bases, and—because it bases its choices on tests using the current case-bases—automatically reflects the characteristics of the specific case-bases and cross-case-base adaptation strategies.

```

THRESHOLDDISPATCH( $P, k, local\_CB, external\_CB$ )
1   $local\_soln\_dist \leftarrow$  average distance from  $P$  to the  $k$  nearest cases in  $local\_CB$ 
2  if  $local\_soln\_dist \leq dispatch\_threshold$ 
3      then solve  $P$  using  $local\_CB$ 
4  else  $external\_soln\_dist \leftarrow$  avg. dist. from  $P$  to  $k$  nearest cases in  $external\_CB$ 
5      if  $local\_soln\_dist \leq external\_soln\_dist$ 
6          then solve  $P$  using  $local\_CB$ 
7          else solve  $P$  using  $external\_CB$ 

```

Algorithm 2. Distance-threshold-based case dispatching algorithm.

Threshold-Based Dispatching: In threshold-based dispatching, a fixed threshold $T \geq 0$ determines the maximum acceptable distance between new problems and the problems in stored local cases. A problem P is dispatched to the external case-base if and only if both (1) P is judged too dissimilar to the problems solved by local cases, and (2) the external case-base contains more similar cases than the local case-base. Algorithm 2 summarizes our dispatching method. Because it is designed for systems using k-NN retrieval, it considers the average similarity between P and the k most similar cases.

Calibrating threshold-based dispatching: Our method for selecting a dispatching threshold examines a test set of problems (e.g., the first problems in the problem-stream), and selects a threshold value to maximize a pre-defined function assessing the utility of solutions. If the goal is simply to maximize the number of problems solved correctly, the utility function assigns a value of 1 if the problem is solved correctly, and 0 if it is solved incorrectly. However, the utility function could also reflect factors such as percent error, solution generation time, bandwidth use, fees to access a particular case-base, etc.

The calibration algorithm first sorts the test problems p_i in ascending order according to their average distance d_{p_i} from the nearest case in the local case-base. The midpoints between d_{p_i} values for adjacent problems are taken as candidate dispatching thresholds; two additional possible thresholds are 0 and infinity. The choice of the threshold can be seen as dividing this ordered set: When a threshold is chosen, all problems with smaller d_{p_i} values will be processed locally, and all other problems will be dispatched. A threshold of zero results in all problems being dispatched; a threshold of infinity corresponds to all problems being solved locally.

The calibration algorithm computes the solution for each test problem in both the local and external case-base and applies the utility function to determine the utility of the resulting solution. It chooses the threshold T to maximize the sum of the utility function values when each problem is solved in the case-base determined by the threshold. In the following experiments, our utility function measures prediction performance.

Case-Based Case Dispatching and Calibration: In our case-based case dispatching method, a *dispatching case-base* is generated by solving the test problems in different case-bases and recording the utilities of the solutions from those case-bases. The dispatching case-base is then used to predict the performance of each candidate case-base (local or external) for similar future problems. Given a new problem to solve, the lo-

cal or external case-base to use is determined by a weighted voting scheme, in which each problem in the dispatching case-base votes for the local case-base or an external case-base, whichever solved it most effectively, and the votes are weighted according to the similarity of each problem in the dispatching case-base to the new problem to be solved. The dispatching case-base is built by solving each test problem in the local case-base and in each external case-base, and storing cases containing the test problem, case-base identifier, and utility value.

When an input problem is received during normal processing, the problem is dispatched to the case-base (local or external) with highest expected utility, according to the following formula. Let P be an input problem, let k -reference_probs(P) be the k nearest neighbors of P in the dispatching case-base, and $soln_utility(R, CB)$ be the utility of solving reference problem R using case-base CB . The expected utility of using case-base CB_i to solve P is calculated by:

$$CB_utility(CB_i, P) = \frac{1}{k} \sum_{R \in k\text{-reference_probs}(P)} soln_utility(R, CB_i).$$

P is dispatched to the case-base CB_i that maximizes $CB_utility(CB_i, P)$.

4.3 Combining Dispatching with Selection of Cross-Case-Base Adaptation Strategies

A variant of the threshold-based calibration algorithm can be used to select both a dispatching method and a cross-case-base adaptation strategy. In that method, the calibration process applies each cross-case-base adaptation strategy to each problem in the test set, and sets the threshold and cross-case-base adaptation procedure to the combination that maximizes the sum of the utilities for the test problems.

Because the case-based dispatching method makes its choices based on performance on similar problems, it can also easily be extended to simultaneously select the external case-base and cross-case-base adaptation strategy most applicable to a particular type of problem. To build the dispatching case-base, each test problem is solved using all combinations of case-bases and adaptation strategies, with the best case-base and cross-case-base adaptation strategy stored in the dispatching case-base. When the system processes a new problem, the voting procedure then votes for pairs (case-base, cross-case-base adaptation strategy). The application of this type of approach to federations of case-bases is discussed in [6].

5 Experimental Illustrations

We conducted experiments both to study the effects of case dispatching and cross-case-base adaptation methods, and to explore the calibration strategies of the previous section. Our experiments explored the following questions:

1. How do dispatching threshold and cross-case-base adaptation strategies affect system performance? This provides a baseline illustration, replicating results from [5] using a different case-base.
2. Can threshold-based dispatching with a small calibration set select the right cross-case-base adaptation strategy?

3. Can case-based dispatching with a small calibration set predict the right external case-base and cross-case-base adaptation strategy?
4. How does the size of the calibration problem set affect the performance of case-based dispatching and selection of cross-case-base adaptation strategies?

Questions (1) and (2) address dispatching to a single external case-base; (3) and (4) consider dispatching to a set of external case-bases.

5.1 Experimental Setup

The task for our experiments was predicting median housing prices, using a publicly-available data set published on the web by the Delve group (Data for Evaluating Learning in Valid Experiments).³ This data set includes 22,784 cases from the 1990 U.S. census, divided by states. The experimental task was predicting prices from partial case-bases for one state, augmented by drawing on the complete case-base for another state (or states) as the external case-base(s). Here all cases use the same representation scheme, but the price for a given set of property features will change based on differences in the housing markets in different states. Intuitively, this prediction task can be seen as related to what a real estate appraiser might do after moving to a new area, when it is necessary to reason from a combination of local and non-local experience. The state case-bases used (with their abbreviations and sizes) are Alabama (AL, 470 cases), Arkansas (AK, 471 cases), Florida (FL, 752 cases), Indiana (IN, 590 cases), Illinois (IL, 1308 cases), Kentucky (KY, 471 cases), Ohio (OH, 1051 cases) Mississippi (MI, 500 cases), and Nevada (NV, 56 cases).

We expect multi-case-base reasoning to be most useful early on, when the reasoner has a limited local case library. To model the stages in the growth of a case-base, sparse versions of the local case-bases were generated by randomly selecting varying percentages of the original local case-base. In what follows, the initial case-bases are designated by CB_1 , and the sparse versions of these case-bases as CB_1^* . Prediction performance is evaluated using all problems in the full CB_1 case-base as test problems (except those used as test problems for selecting the dispatching threshold or for case-based dispatching), with leave-one-out cross validation. Performance is measured by counting the number of cases whose prices were predicted correctly to within an error margin of 20%. All predictions are done using a distance weighted 3-nearest-neighbor algorithm (3-NN). Note that no effort was made to tune feature weights to achieve better performance; the goal was to study the comparative value of MCBR, rather than to maximize performance.

5.2 How do dispatching threshold and cross-case-base adaptation strategies affect system performance?

In this experiment, the system determined whether to handle the case locally within CB_1^* or to dispatch it to an external case-base CB_2 , based on the problem's similarity to cases in CB_1^* and CB_2 and on the dispatching threshold. If the problem was

³ <http://www.cs.toronto.edu/~delve/data/census-house/desc.html>.

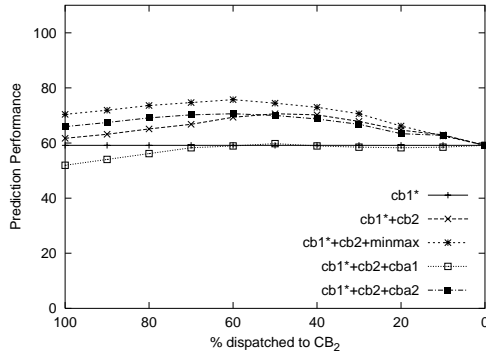


Fig. 1. Prediction accuracy for AL housing prices, using a sparse case-base of AL census data supplemented with dispatching to a full case-base of KY prices, with cross-case-base adaptation.

dispatched, cross-case-base adaptation was applied to the predicted value from CB_2 before evaluating its accuracy. Figure 1 illustrates the impact of dispatching criteria and choice of adaptation strategies on performance. Here the local and external case-bases both reflect somewhat similar tasks (predicting house prices in AL and KY), but with very different competences—the local case-base of AL cases is very sparse. CB_1^* contains only 5 cases, drawn randomly from census data for AL, compared to the full KY case-base CB_2 .

Figure 1 shows prediction accuracies of the four different cross-case-base adaptation strategies from Section 4.1 (ID, Minmax, CBA1 and CBA2). Identity cross-case-base adaptation is indicated by $CB_1^* + CB_2$; linear adaptation based on global min and max is indicated by $CB_1^* + CB_2 + \text{minmax}$; the two local adaptation methods are indicated by $CB_1^* + CB_2 + \text{cba1}$ and $CB_1^* + CB_2 + \text{cba2}$, respectively. The horizontal line, marked CB_1^* , shows baseline performance with no case dispatching. For other conditions, problems are dispatched when no “sufficiently similar” cases are available in CB_1^* , following Algorithm 2.

These experiments were run for a wide range of sparse case-bases. When the distance threshold is too low, problems are dispatched when they could be solved locally, and performance may be degraded by flaws in cross-case-base adaptation. When the threshold is too high, cases are solved locally even when they are outside of the local case-base’s competence. However, for the right choice of threshold, as in the 60% dispatched condition of the graph, performance noticeably surpasses the local case-base alone. This pattern appears primarily in sparse case-bases—in our experiments, dispatching had less benefit as the density of the local case-bases increased.

We note that performance depends on both the dispatching threshold and the cross-case base adaptation strategies chosen. For this sample data, performance is actually worse with CBA1 adaptation than if problems are dispatched with no cross-case-base adaptation at all. This example shows that the choice of appropriate strategies is crucial to the success of MCBR.

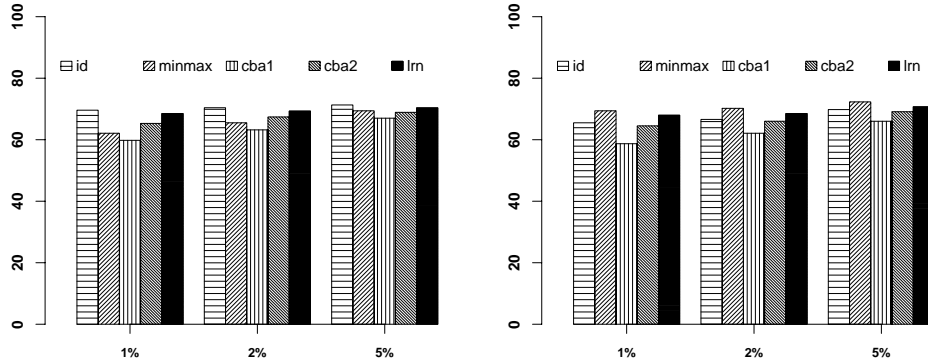


Fig. 2. Left: Predicting AL median house prices, using a sparse AL case-base as the local case-base, with an external case-base of MI census data. Right: Predicting AL prices, using a sparse AL case-base as the local case-base and an external case-base of KY census data.

5.3 Can threshold-based dispatching with a small calibration set predict the right cross-case-base adaptation strategy?

To explore the feasibility of automatically selecting strategies to tune case dispatching and cross-case-base adaptation, we performed a series of experiments, the results of some of which are shown in Figure 2. These show two samples of the effects of the calibration process for selecting the cross-case-base adaptation strategy to use for two different external case-bases, MI and KY. For both graphs, the local case-bases are randomly-selected subsets of the AL case-base. Tests were run for subsets containing 1%, 2%, and 5% of the AL cases, to simulate effects as the local case-base grows over time. For each density, the first bar shows predictive accuracy with no cross-case-base adaptation (ID), the second with minmax, the third and fourth with CBA₁ and CBA₂, and the last (LRN) with the learned method selected by the threshold-based calibration process described in Section 4.2. Graphs represent the average of 10 runs, each using a 10-problem test set for calibration and the remaining 460 AL cases for testing.

For both examples, the case-bases are sufficiently similar that the identity cross-case-base adaptation gives good performance. However, better performance is sometimes achieved with other methods, and the best methods vary. For example, for the MI external case-base and 1% local case-base, the identity outperforms all other methods, and is noticeably superior to minmax. For the KY external case-base and 1% local case-base, minmax outperforms the identity. Thus, selecting a single fixed strategy will sometimes give poor performance, and the question is whether calibration can consistently give performance close to that of the optimal strategy, whichever that strategy may be. In these experiments, calibration resulted in performance close to that of the optimal strategy. Although these choices were surpassed on individual runs, calibration resulted in better average performance than any fixed strategy, even though calibration was based on only 10 test problems.

5.4 Can case-based dispatching with a small calibration set predict the right external case base and cross-case-base adaptation strategy?

The overall goal of MCBR with self-calibration is to automatically adjust the MCBR system's dispatching and cross-case-base adaptation to make the best use of multiple case-bases. To test this ability, we compared the performance of a variety of strategies: using the local case-base only, using one of 12 fixed combinations of external case-bases and strategies (3 external case-bases \times 4 cross-case-base adaptation strategies), using randomly-chosen combinations of case-bases and cross-case-base adaptation strategies, and using the case-based method of Section 4.2 to learn to select case-bases and cross-case-base adaptation strategies from the calibration process. All runs used sparse versions of the Indiana case-base (IN*) as the local case-base, and test problems drawn from the full IN case-base, with leave-one-out cross-validation. One set of runs used MCBR with the external case-bases IL, KY, and OH, all of which were expected to have property valuation characteristics similar to the local case-base. Another used the external case-bases FL, IL, and NV, in which two case-bases, FL and NV, were expected to have quite different characteristics. 10 randomly-selected versions of the local case-base were tested as local case-bases, with 5 runs for each one using different randomly-selected calibration sets, for each of the four densities 1%, 2%, 5%, and 10%, for a total of 200 test runs.

As expected, the average benefits of MCBR were greatest for sparse local case-bases. Figure 3 shows average results for the sparsest local case-base tested (1%). The left-hand graph shows results when the local case-base and external case-bases make similar predictions (in fact, the IL case-base with no cross-case-base adaptation outperforms the sparse local case-base IN*). We note that for each of the external case-bases, there is a cross-case-base adaptation strategy for which dispatching markedly outperforms the sparse local case-base. On average, even random dispatching and selection of adaptation outperform the sparse local case-base, but this is outperformed by the learned strategy, whose performance is close to the maximum observed.

In the right-hand graph, the cases in the FL and NV case-bases are dissimilar to those in IN*, and the cross-case-base adaptation strategies are insufficient to compensate. Here random dispatching is much worse, but performance with learned dispatching again surpasses the local case-base and is close to that of the best external case-base and cross-case-base adaptation strategy. In our tests, the learned method consistently approximated or surpassed the optimal single-method performance. As the local case-base became less sparse, the advantage of dispatching decreased, but learning still approximated the best performance, by sending problems to the local case-base. Thus the calibration method appears to be robust enough not only to guide dispatching when external case-bases are helpful, but also to shift to the local case-base when appropriate.

5.5 How does the size of the calibration set affect performance of case-based dispatching and selection of cross-case-base adaptation strategies?

The benefit of tuning strategies will depend on the sufficiency of the calibration set to guide dispatching and choice of cross-case-base adaptation methods. Ideally, a small set

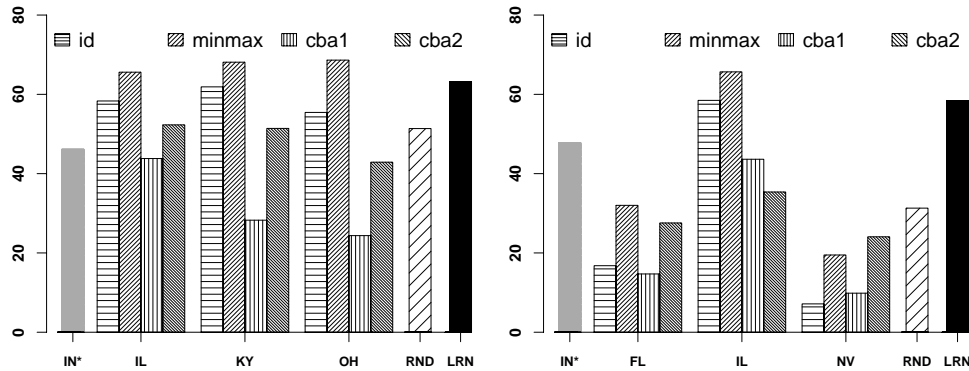


Fig. 3. . Comparison of performance of a sparse local case-base (IN*) with two sets of external case-bases (IL, KY, OH, left, and FL, IL, NV right) for four cross-case-base adaptation strategies. RND designates random dispatching and cross-case-base adaptation; LRN designates the choices learned from the test problems by building up the dispatching case-base.

would be sufficient, both to minimize calibration effort and to decrease the cost of case-based dispatching. However, we expect smaller sample sizes to degrade performance. Thus an important question is how the calibration set size affects performance: whether the methods can make good dispatching decisions with limited-size samples.

To explore the effects of sample size on system performance, we performed an experiment to compare the performance of case-based dispatching/cross-case-base adaptation selection for different calibration set sizes. As the local case-base, we used a 1% version of the sparse Indiana case-base (IN*). Calibration sets of all sizes ranging from 1–30 cases were randomly selected from the IN case-base, resulting in calibration set sizes ranging from 0.2%–5% of the total problem set. Additional calibration sets were generated up to size 200, in steps of 10. Performance was tested on all remaining cases in the original IN case-base, by leave-one-out cross validation. Runs were performed 15 times, selecting random subsets of IN for calibration, and the results were averaged.

Figure 4 compares the performance achieved with these sample sets to the performance that would be achieved by case-based selection of dispatching and cross-case-base adaptation, if the entire problem set were available for calibration. The figure compares five scenarios: (1) standard CBR with no dispatching, (2) calibration used to guide MCBR (dispatching and cross-case-base adaptation), to draw on three external case-bases expected to have similar characteristics to the local case-base (IL, KY, and OH), (3) random choice of case-bases between the local case-base IN* and the IL, KY, and OH case-bases, with random choice of cross-case base adaptation strategies when external case-bases are used, (4) calibration used to guide MCBR (dispatching and cross-case-base adaptation) to draw on three external case-bases including two case-bases expected to have characteristics divergent from the local case-base (IL, FL, and NV), and (5) random choices, as in (3), with the external case-bases IL, FL, and NV.

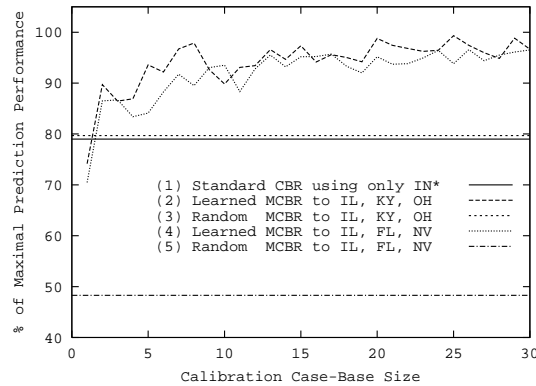


Fig. 4. The effects of calibration sample size on performance, with performance expressed as a percentage of the performance obtained when the entire case-base is used for calibration.

For a single case in the calibration set, performance is noticeably below that with the local case-base alone. Surprisingly, calibration on only two cases was sufficient to surpass the local case base on average. As expected, there is an increasing performance trend as the size of the calibration set increases. With 30 cases in the calibration set, performance was 96% of the performance achieved by calibrating on the entire case base, with the improvement curve leveling out for larger calibration set sizes.

When the external case-bases are similar to the local case-base (for external case-bases IL, KY, OH), the average performance of the local case base alone (line 1) is similar to that achieved by random dispatching (line 3). However, when some of the external case-bases are quite different (IL, FL, NV), random dispatching results in poor performance (line 5). Either way, the learned strategies (lines 2 and 4) markedly outperform both the local case-base alone and random dispatching. These results are encouraging both for the value of learning MCBR strategies and for the ability to do so from limited sample sets.

6 Perspective

Relationship to distributed CBR: Previous research has proposed approaches in which multiple CBR agents share contents of their individual case-bases as needed [11, 7, 10] and have studied fundamental issues for efficient access of distributed case-bases [4], with the assumption that the case-bases are standardized. Recent research has also observed the benefits of strategically accessing case-bases with specialized coverage [8]. *Multi-case-base reasoning* differs in addressing questions of applying case-bases that may have been accumulated in other contexts, for somewhat different tasks, and consequently involve tradeoffs.

Most relevant to this paper is Ontañón and Plaza's [9] work on collaborative case-based reasoning. Their Bounded Counsel Policy uses a *termination check* to determine when a local agent's solution is unreliable, prompting consultation with other agents; they also develop methods for learning the termination check. Differences include that

termination checks form judgments based on solutions, rather than problems, and that their focus is on deciding when no more solutions need be sought from external case-bases, rather than on learning which external case-bases are the best to consult.

Relationship to Case-Base Maintenance: The self-tuning approach to adjusting dispatching and cross-case-base adaptation can be seen as a form of *case-based reasoner maintenance* [12]. Case-based reasoner maintenance goes beyond maintaining the case-base, to automatically adjust other aspects of the CBR system. The calibration process proposed here for cross-case-base adaptation, for example, could be applied to select adaptation strategies in a single-case-base system. Although prior CBR systems have learned to refine certain characteristics (e.g., their indexing criteria), to our knowledge, our project is the first effort to develop CBR systems that can choose from a repertoire of processing strategies based on current circumstances. We consider work in this area to be a promising future direction for developing self-maintaining CBR systems that can dynamically adjust themselves to new circumstances.

Relationship to Distributed Databases and KDD: Research in IR, KDD, and distributed databases is pursuing a number of areas that may prove useful to MCBR. For example, the question of how large a sample to use to predict data streams, which may be important in selecting a test set for self-calibration, has been studied in [3]. In the IR community, strategies have been developed for estimating the contents of data sets with small samples of their data [1]. Analogous strategies may be useful for making more informed choices about which case-bases to access—by estimating their areas of competence—as well as for choosing test sets of cases for calibration.

An important issue beyond the scope of this paper is how to establish correspondences between case representations, if the representations used by different case-bases differ. Recent databases research on enabling automatic transformation of XML documents, and on learning to map between structured representations (e.g., [2]) promises to be useful for this task.

7 Conclusions

When external case-bases are available to supplement local case knowledge, they can provide a valuable additional resource, especially during the early stages of the growth of a case-base. However, unless the external case-base addresses precisely the same task, cases from external case-bases may not be immediately applicable, necessitating cross-case-base adaptation. Unfortunately, it may not be clear which external case-bases to draw on, which cross-case-base adaptation strategies to apply, or whether the results of applying those strategies will be adequate. The benefits of MCBR depend strongly on the specifics of the problem at hand and how increased case similarity balances against differences in the tasks and environments for which different case-bases were gathered. Unless it is possible to determine when and how to apply MCBR, there is no guarantee that MCBR will be advantageous.

This paper has presented methods that enable an MCBR system to learn when and how to draw on external case-bases, based on testing a small sample of domain prob-

lems. Given a set of possible strategies and a utility function to compare alternative solutions, these methods select the strategies that give the best performance. The calibration methods may be applied to any task for which the utility of solutions can be evaluated; the paper has presented experiments on applying self-calibration to choose between knowledge-light strategies for a numerical prediction task, showing that simple methods can be sufficient to achieve good performance. For our test examples, performance of learned strategies far exceeds the performance of local sparse case-bases alone, and performance with learned strategies is close to the performance achieved by the optimal external case-base and cross-case base adaptation strategies. This establishes the promise of the approach.

More broadly, an interesting area for future research is the application of analogous methods to enable traditional CBR systems to tune themselves by testing which strategies work best for samples of the problems that they encounter. Thus we believe that the current results are promising not only for self-tuning in multi-case-base reasoning, but as a step towards CBR systems that can adjust themselves to their tasks.

References

1. J. Callan. Query-based sampling of text databases. *ACM Transactions on Information Systems*, 19(2):97–130, 2001.
2. A. Doan, P. Domingos, and A. Halevy. Reconciling schemas of disparate data sources: A machine learning approach. In *Proceedings of the ACM SIGMOD Conf. on Management of Data (SIGMOD-2001)*, Menlo Park, 2001. ACM Press.
3. P. Domingos and G. Hulten. Mining high-speed data streams. In *Knowledge Discovery and Data Mining*, pages 71–80, 2000.
4. Conor Hayes, Pádraig Cunningham, and Michelle Doyle. Distributed CBR using XML. In *Proceedings of the KI-98 Workshop on Intelligent Systems and Electronic Commerce*, 1998.
5. D. Leake and R. Sooriamurthi. When two case bases are better than one: Exploiting multiple case bases. In *Proceedings of the Fourth International Conference on Case-Based Reasoning, ICCBR-01*, Berlin, 2001. Springer-Verlag.
6. D. Leake and R. Sooriamurthi. Managing multiple case-bases: Dimensions and issues. In *Proceedings of the Fifteenth FLAIRS Conference*, pages 106–110, Menlo Park, 2002. AAAI Press.
7. F. Martin, E. Plaza, and J.-L. Arcos. Knowledge and experience reuse through communications among competent (peer) agents. *International Journal of Software Engineering and Knowledge Engineering*, 9(3):319–341, 1999.
8. L. McGinty and B. Smyth. Collaborative case-based reasoning: Applications in personalised route planning. In *Proceedings of the Fourth International Conference on Case-Based Reasoning*, Berlin, 2001. Springer Verlag.
9. S. Ontañón and E. Plaza. Learning when to collaborate among learning agents. In *Machine Learning: ECML 2001*, pages 395–405, Berlin, 2001. Springer-Verlag.
10. E. Plaza and S. Ontañón. Ensemble case-based reasoning: Collaboration policies for multi-agent cooperative CBR. In *Proceedings of the Fourth International Conference on Case-Based Reasoning, ICCBR-01*, Berlin, 2001. Springer-Verlag.
11. M. V. Nagendra Prasad, V. Lesser, and S. Lander. Reasoning and retrieval in distributed case bases. *Journal of Visual Communication and Image Representation*, 7(1):74–87, 1996.
12. D. Wilson and D. Leake. Maintaining case-based reasoners: Dimensions and directions. *Computational Intelligence*, 17(2):196–213, 2001.