International Journal on Artificial Intelligence Tools: Architectures, Languages and Algorithms (IJAIT). Special issue on recent advances in techniques for intelligent systems. Vol 13, No 1, pp 237–254.

International Journal on Artificial Intelligence Tools

CASE DISPATCHING VERSUS CASE-BASE MERGING: WHEN MCBR MATTERS

DAVID B. LEAKE

Computer Science Department, Lindley Hall 215
Indiana University
150 S. Woodlawn Avenue
Bloomington, Indiana 47405, U.S.A.

RAJA SOORIAMURTHI

Kelley School of Business, Business 540D
Indiana University
1309 East 10th Street
Bloomington, Indiana 47405, U.S.A.

Received (July 18, 2003)
Revised (December 7, 2003)

Multi-case-base reasoning (MCBR) extends case-based reasoning to draw on multiple case bases that may address somewhat different tasks. In MCBR, an agent selectively supplements its own case-base as needed, by dispatching problems to external case-bases and using cross-case-base adaptation to adjust their solutions for inter-case-base differences. MCBR has been advocated as a means to facilitate handling large case-bases when storage is limited, or to enable use of distributed case sources. However, this raises an important question: When storage is not an issue, and the cases from all external case sources could be merged into a single case-base, is there any reason for MCBR? This article answers that question with an experimental assessment of how MCBR affects the quality of solutions generated. It demonstrates that for a given local case-base and an external case-base for a task environment that is similar to, but different from, the local task environment, MCBR can improve accuracy compared to merging the case-bases into a single case-base. This improvement holds even if the cross-case-base adaptation method used by MCBR is also applied to the external cases before merging. The article hypothesizes an explanation of this behavior in terms of the ability of MCBR to exploit the tradeoffs between similarity of problems and similarity of solution contexts. It provides experimental evidence to support this hypothesis, and also demonstrates that MCBR is a useful framework for guiding case-base maintenance by selecting cases to add to a case-base.

Keywords: Case-based reasoning, case-base maintenance, case discovery, case dispatching, cross-case-base adaptation, distributed systems, multi-case-base reasoning.

1. Introduction

Case-based reasoning (CBR) systems reason from experience, solving new problems by retrieving solutions to similar prior problems from a case-base of experiences, and adapting their solutions to fit new needs.[1,2,3,4] Multi-case-base reasoning (MCBR) addresses how case-based reasoning systems can supplement their own case-bases by drawing on the experiences of other case-based reasoners: how they can make effective use of external case-bases which may have been generated for related, but possibly non-identical tasks.[5,6,7] When the local case-base is sparse, as in the early phases of case-base development, MCBR effectively extends the system's case-base by importing cases. MCBR reasons about issues such as which case-base is most suitable for solving the current problem and how to revise solutions in light of inter-case-base differences. Using MCBR to access external case sources as needed contrasts with "eager" merging, in which all cases from all sources are standardized and merged into a combined case-base.

MCBR has been advocated as a means to facilitate processing of large case-bases, by handling subsets of the case-base separately, and for exploiting distributed case sources which may not be available to merge. However, this raises an obvious question: When merging is practical, is there any reason to perform MCBR rather than simply merging all cases into a single case-base? This article examines the question of when MCBR is preferable to case-base merging, focusing on an issue that has not been studied previously: The comparative effects of MCBR and eager merging on solution accuracy. The article presents experiments demonstrating a surprising result: that when local and external case-bases reflect somewhat different task circumstances, MCBR can result in markedly higher quality solutions than eager merging, even when the same cross-case-base adaptation strategy is applied to all external cases. The article hypothesizes an explanation for this behavior—that MCBR enables making useful tradeoffs between similarity and expected solution quality when using cases from different sources. It then provides evidence for this hypothesis with an additional experiment showing that its predictions are borne out in another context, when the local and external case-bases reflect identical tasks but external cases have noisy solutions. It also demonstrates that MCBR processes can be useful in case-base maintenance/case discovery, providing an effective strategy for selectively adding cases to the case-base. These results show that MCBR is sometimes useful not only for space and case availability concerns, but for making the best use of external case resources.

2. The MCBR Process

In the standard CBR process, new problems are solved by retrieving cases for similar prior problems from a case-base of prior experiences, adapting their solutions to enable them to be reused in the new circumstances, revising the results as needed if the generated solution does not apply, and retaining the new case for future use.[1,2,3,4] MCBR augments this process (1) by enabling the reasoner to choose to retrieve prior cases from any of a number of case-bases, and (2) by adapting suggested solutions based not only on the new problem circumstances, but also on

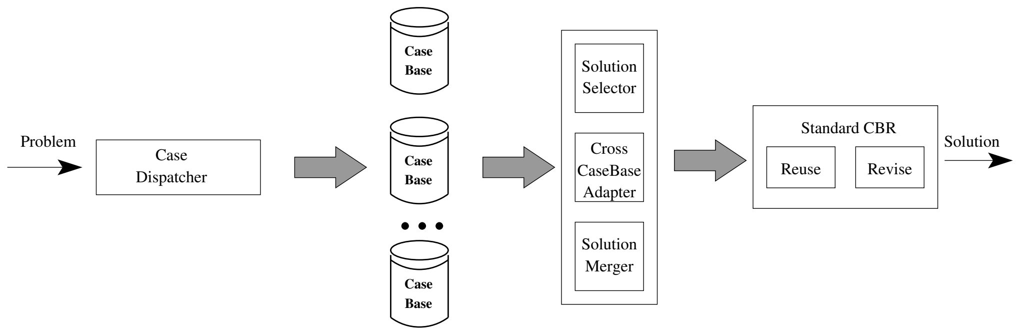


Figure 1: Multi-case-base reasoning framework for drawing on a set of case-bases, from Leake and Sooriamurthi (2002).[6]

information about the characteristics of the case-bases from which they were drawn.

Figure 1 illustrates one possible MCBR architecture. When problems are input to the system, a dispatcher selects case-bases to query and a strategy for pursuing the query sequence. As results are returned, the system then selects which of the returned cases to consider, performs cross-case-base adaptation for inter-case-base differences, and merges solutions if needed. These steps may be ordered in different ways, e.g., cross-case-base adaptation may precede or follow the solution merging step.

For example, in an e-commerce application, a local merchant asked to estimate the price for a particular home theater system might first seek product information from its own case-base, and, if no sufficiently similar cases were available, might dispatch the queries to other suppliers. When their cases for similar products are returned, their prices may require revisions based on systematic inter-case-base differences (e.g., to convert prices from Euros to dollars, if cases from a European supplier were being retrieved for American use, or to increase the estimate, if the other supplier sells at a discount that the local merchant cannot match). Another task suitable for MCBR is case-based spam filtering, in which prior examples are used to determine how to classify new messages.[8] Because different users have different preferences, it is desirable to keep a distinct case-base of examples for each user; merging all users' case-bases would blend individual preferences and potentially increase errors. However, when none of a user's own cases are sufficiently close to a new message, drawing on other users' case-bases is useful.

The general MCBR framework may treat the local case-base in a special way, or may treat all case-bases, local and external, according to a uniform set of dispatching criteria. For example, if the local case-base has a low access cost, a system might only dispatch problems if the local case-base lacks a similar solution; alternatively, if solution quality is paramount, it might treat all case-bases equally according to a quality-oriented dispatching strategy. In what follows, for simplicity we will assume that there is a single fixed local case-base and a single fixed external case-base.

However, tests from our previous research suggest that similar results would apply for larger numbers of external case-bases.[7]

3. Relationship of MCBR to Previous Research

The general idea of multi-case-base reasoning relates to research in a number of areas, both within case-based reasoning (e.g., on distributed CBR) and in research areas such as heterogeneous and distributed databases. Some of these relationships are described in Leake and Sooriamurthi (2002).[7]

Previous CBR research has studied the use of sets of cases at different levels of abstraction, to solve problems hierarchically.[9,10] MCBR, in contrast, focuses on issues arising when a system may draw on cases from multiple case-bases at a single level, with each addressing a similar problem at the same level of abstraction, but perhaps for different circumstances. Thus how to choose between competing case-bases is a central issue for MCBR. The MCBR dispatching process can also be seen as defining a new similarity measure, applied to the union of all case-bases, based on expected utility of accessing cases from different case-bases. Thus one of the claims of MCBR is that combining learned dispatching criteria with the case-bases' own similarity metrics is a convenient "knowledge light" way to reflect utility differences between cases in different case-bases.

McGinty and Smyth's (2001) work on collaborative CBR[11] is in the most similar spirit, in addressing how to draw on case-bases from other agents who have performed the same type of task, but whose preferences may differ, and in raising the question of the tradeoff between case-base similarity and coverage (cf. Leake and Sooriamurthi, 2001).[5] McGinty and Smyth's dispatching method compares the contents of different case-bases to identify agents with similar preference models, in order to select useful cases. However, instead of attempting to select a single case-base to query, as in MCBR, their method broadcasts a query to all case-bases, and selects a result from that complete set of returned results. Also, MCBR adds a new adaptation step: in addition to normal CBR case adaptation, the differences between case-bases are used to determine cross-case-base adaptation strategies, to adjust cases for inter-case-base differences.

Ontañón and Plaza (2001, 2003)[12,13] also study the exploitation of solutions generated by different agents. However, their work focuses on determining when a sufficiently good solution has been obtained, rather than on deciding which case-base to query, and in focusing on when an agent should accept a case offered by an external source, rather than on when to take the initiative to request another case to augment the local case library.

The considerations of trade-offs between problem and context similarity in MCBR also relate to research on ensemble reasoning methods for CBR (e.g., Cunningham and Zenobi, 2001).[14] Likewise, improving performance by choosing between multiple case-bases and adaptation strategies relates to machine learning research on committee-based approaches and methods such as supra-classifiers (e.g., Bollacker and Ghosh, 1998).[15]

The MCBR case dispatching and selection process can be seen as corresponding to defining a new similarity metric, applicable over the union of case-bases, which reflects quality concerns and determines which cases to use in a new situation. However, it should be noted that the criteria used by MCBR are not necessarily restricted to case similarity or solution quality—e.g., they may reflect factors such as access times and costs for drawing cases from different external case-bases—and that they explicitly reflect the original boundaries between case-bases in their decision-making, in order to reflect the task and environment characteristics surrounding those case-bases.

A significant body of CBR research studies case-base maintenance (see Leake, Smyth, Yang, and Wilson, 2001, for a sampling of recent research).[16] Maintenance approaches normally focus on tuning the case-base to improve expected future performance, for example, by building case-bases that are compact but maintain competence over the space of potential problems.[17] Case discovery processes can then be aimed at filling gaps in the system's current coverage.[18] In contrast to this "eager" approach of filling gaps to prepare for possible future problems, MCBR can be seen as a new maintenance approach for performing focused "lazy" case addition: MCBR imports cases, and converts them to local requirements by cross-case-base adaptation, as needed to deal with specific new problems. An alternative maintenance strategy would be to simply merge all case-bases, with adaptation for inter-case-base differences. This article presents the perhaps-surprising result that when case sources have different characteristics, using MCBR to decide when to access cases from different case-bases as needed can achieve better solution quality than using a single case-base, even when all the cases have been adapted for inter-case-base differences.

Cross-case-base adaptation may be seen as related to databases research on enabling automatic transformation of XML documents, and on learning to map between structured representations (e.g., Doan, Domingos and Halevy 2001),[19] and on semantic integration in general.[20] However, that work addresses issues of drawing on multiple data sources and accounting for their representational differences, while our current work focuses on content differences for a shared representation.

4. Learning when to Dispatch Cases and to Select Cross-Case-Base Adaptation Strategies

A major focus of our work has been on determining when to draw on cases from external sources, and how to adapt them in response to inter-case-base differences. We have explored a dispatching method in which problems are dispatched to an external case-base if their distance from the most similar local case exceeds a threshold, as well as methods for automatically setting this threshold.[7] The dispatching method treats the external case-base and its similarity/retrieval method as a "black box," analyzing system performance on initial problems and selecting the dispatching threshold that gives the best performance on that sample.

We have explored two case dispatching strategies, *threshold-based dispatching*[5] and *case-based dispatching*.[7] Threshold-based dispatching dispatches retrieval queries to an external case-base if local cases are not available for sufficiently similar problems, based on a similarity threshold selected for the local and external case-bases and problem stream. Case-based dispatching dispatches problems to the external case-bases that have provided the best performance on similar previous problems. The performance metric can consider not only solution quality, but also factors such as communications costs, bandwidth constraints, case-base access fees, etc.

For both dispatching methods, we have investigated the effectiveness of learning criteria for when to dispatch. Our learning methods examine the performance of all the available case-bases on a sampling of problems (which we call the calibration set), drawn from the beginning of the problem stream to be processed by the system. For case-based dispatching, we have developed an algorithm that selects a threshold that optimizes performance on the calibration set. Likewise, our work on learning to adjust MCBR includes studies of using the same calibration data to select cross-case-base adaptation strategies from a standard repertoire (currently, a set of knowledge-light, domain-independent strategies developed for regression tasks). In our tests, these approaches give good performance in practice; full details are available in Leake & Sooriamurthi (2002).[7]

A particular benefit of the learning methods is that they address the problem of deciding when to use MCBR: If drawing on an external case-base is unhelpful to system performance (e.g., because of substantial differences in the tasks, task environments, or even similarity metrics), the system learns to rely on the local case-base.

5. New Motivations for MCBR: Two Hypotheses

A number of potential benefits have been advanced for using MCBR[6]. These include increasing efficiency—dividing up a large case-base into sub-case-bases processed individually may increase retrieval speed—augmenting coverage when needed, and exploiting distributed case information available on a per-case basis. In addition, in some cases, privacy concerns or communication costs for transmitting case-bases may dictate against forming a single unified case-base, or the owners of distinct cases-bases may not be willing to provide their entire case-bases, only making individual cases available. Likewise, when numerous individual resources exist without central control, it may be more practical to draw on individual information sources than to attempt to generate a single uniform repository.

Although there are compelling arguments for these advantages, they apply primarily when constraints on processing speed, storage, or case access prevent merging the different case-bases. If those constraints do not apply, it would be possible to simply import all external cases as a group, perform cross-case-base adaptation on the entire group, and then to reason from the single resulting case-base by the normal CBR process. However, this article proposes two new motivations that can make MCBR useful even when it is practical to perform eager case-base merging:

1. **Increasing solution quality:** If two case-bases are merged, all the cases in the resulting case-base are treated uniformly for future retrievals, and the criterion for choosing between their cases is simply their similarity to new problems. In MCBR, however, considerations beyond similarity—such as the quality of cross-case-base adaptation—can be traded off against similarity. This prompts the MCBR quality hypothesis:

   > *The MCBR quality hypothesis*: MCBR—lazy retrieval and cross-case-base adaptation—should sometimes provide improved solution quality compared to cross-case-base adapting all external cases and merging them with the local case-base.

2. **Guiding selective case addition:** When a CBR system has insufficient competence, additional cases may be needed. Research on case discovery examines how to select cases to add[21] and to identify cases to fill competence gaps in the case-base.[18] The MCBR process of dispatching cases to external case-bases may be seen as another strategy for case-base building, in which cases are added to fill only those competence gaps that affect the current performance of the CBR system, given the characteristics of local case-base, external case-base, and cross-case-base adaptation. This prompts the MCBR case-base building hypothesis:

   > *The MCBR case-base building hypothesis*: MCBR is a useful framework for guiding case addition.

These hypotheses are examined in a set of experiments described in the following sections.

6. Experimental Design

Our experiments studied MCBR in the context of predicting median housing prices, using a publicly-available data set from the Delve group.[1] This data set includes 22,784 cases from the 1990 U.S. census, divided by states. In previous experiments, we have also examined the performance of MCBR using the `ai-cbr` travel case-base, a standard public case-base, with similar results for the benefits of MCBR.[5]

Each case in the Delve housing case-base consists of 138 attributes spanning various census statistics such as age, marital status, ethnicity, house occupancy etc. The target task is to predict the median housing unit price. For comparative evaluation purposes the Delve group has partitioned these attributes into 4 standard sets, and our experiments used the attribute set `8L`. Feature values were normalized

[1] http://www.cs.toronto.edu/~delve/data/census-house/desc.html.

according to the guidelines proposed by the Delve group, and all features were weighted equally. Because the goal was to study comparative performance effects, no attempt was made to tune the system for the problem domain.

In the experiments, a randomly-selected subset of the cases from one state was used as the "local" case-base, and the complete case-base for another state was used as the "external" case-base. The reason for including only a subset of the local cases was to simulate the circumstances when MCBR is expected to be useful: when the local case-base has incomplete coverage, so additional cases are needed. To simplify comparison of results, we chose to use only one external case-base. In previous work, we showed the effectiveness of a method for learning to automatically select which case-base to query for a given problem,[7] and consequently, we would expect MCBR performance to remain constant or improve with additional external case-bases, making the single-case results provide a rough lower bound on the performance achievable with additional case-bases. Whether performance for a particular problem would actually improve with more case-bases, however, would depend on their individual characteristics and the effectiveness of the case dispatching process.

All the test case-bases have 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, the price 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), Florida (FL, 752 cases), Indiana (IN, 590 cases), Illinois (IL, 1308 cases), Kentucky (KY, 471 cases), Mississippi (MS, 324 cases), and Ohio (OH, 1051 cases). In the experiments, sparse versions of the IN case-base were used as the local case-base. (An asterisk will designate the sparse version of a case-base, e.g., IN* is the sparse version of the Indiana case-base). Leake and Sooriamurthi (2001),[5] discuss how the benefit of MCBR increases with the sparsity of the local case-base.

For each combination of sparse local case-base and external case-base, prediction quality was compared for lazy and eager merging. Both conditions used test problems from the original IN case-base, with leave-one-out cross validation. For different case-bases, different cross-case-base adaptation strategies will be most suitable. To select the strategy to use, our learning method performed a comparative test of the available methods and selected the method that gave the best performance on the first 30 problems. Four knowledge-light cross-case-base adaptation strategies were compared: identity (no adaptation), linear interpolation (predicted prices were adjusted to fit the range from low to high prices in one case-base to the range from low to high prices in the other), and two alternative case-based adaptation methods, which interpolated based on comparing the prices of nearby previously-solved problems. The algorithm and adaptation strategies are described in detail in Leake and Sooriamurthi (2002).[7]

For the eager merging condition, the merged case-base was generated by taking each case from the external case-base, applying cross-case-base adaptation, and then adding the cross-case-base adapted version to the local case-base. In subsequent tests, normal CBR was done on the resulting case-base, with prices predicted by averaging prices of the 3 nearest neighbors. For MCBR, a dispatching threshold is selected by the learning method described in Leake and Sooriamurthi (2002).[7]

When the distance between an input problem and the most similar case in the local case-base exceeds the threshold, the problem is dispatched to the external case-base. The external case for the most similar problem is then retrieved. If it is more similar than the most similar local case, the solutions of the three nearest neighbors in the external case-base are averaged. Finally, the result of this process is cross-case-base adapted to yield a solution. Prediction accuracy is measured as the percentage of problems for which predicted prices were within 20% of the actual value.

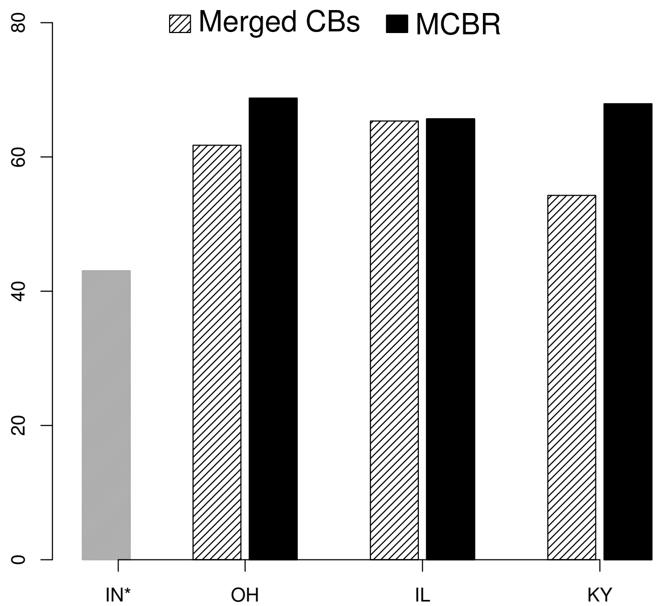
7. Solution Quality Effects for Merging vs. Dispatching

The success of MCBR depends on the availability of external case-bases to fill the local case-base's competence gaps. When such external case-bases are available, merging all case-bases may seem to be a good alternative to MCBR's lazy merging: Eager merging avoids the overhead and potential for error of case dispatching. However, the *MCBR quality hypothesis* suggests that MCBR may be advantageous even when merging is possible. To explore this hypothesis, we performed experiments on the quality effects of eager vs. lazy merging.

7.1. *Experimental Results on MCBR vs. Eager Merging*

Figure 2 graphs the performance of MCBR vs. eager merging for IN* as the local case-base (here IN* contains 1% of the original IN cases), and OH, IL, KY, respectively, as external case-bases. The bars show the average performance for 10 versions of the local case-base, all containing the same number of randomly-selected cases. The first bar shows the prediction performance of the local case-base in isolation. Next, each pair of bars represents the merged and dispatched behavior for MCBR with a given local case-base/external case-base combination. The following table lists averages, standard deviation, and p values for a two-tailed t-test.

In all trials, including tests done with other case-base combinations, the performance of MCBR with case-dispatch is at least as good as that of the merged case-base behavior and often noticeably surpasses it (e.g., for local-external case-base pairs [IN*, KY], [IN*, OH]). Thus under some circumstances, eager merging can be detrimental to the performance of the CBR system. The following section hypothesizes an explanation for the observed behavior.



		OH	IL	KY
μ	Merge	61.74	65.35	54.27
	MCBR	68.77	65.69	67.92
σ	Merge	0.24	0.19	0.69
	MCBR	0.21	0.23	0.15
		$p < 0.0001$	$p < 0.002$	$p < 0.0001$

Figure 2: Performance of eager merging vs. MCBR.

7.2. *Why MCBR May Increase Accuracy*

Initially, it appears surprising that MCBR could improve quality compared to normal CBR after merging all available cases. At the very least, MCBR introduces a new potential source of error, the decision of whether to process a case in the local or (potentially less suitable) external case-base. However, we believe that the benefit of MCBR can be explained by the added control MCBR gives the case selection process, enabling the use of external cases only when they are expected to be beneficial.

The value of using a case from a case-base with somewhat different characteristics depends not only on the similarity of the problem that case solves to the new problem, but also on (1) the level of inter-case-base differences—the extent to which the solutions for similar problems may differ in the local and external case-bases, and (2) the ability of cross-case-base adaptation procedures to compensate for inter-case-base differences. Even if the external case is very similar to the input problem, using that case may degrade performance if it suggests a less-reliable solution than a more dissimilar case from the local case-base.

Figure 3 illustrates a potential bad result of eager merging. In the figure, the x-axis measures the similarity distance of cases to the given target problem. The y-axis measures the quality of the solution associated with a case. Here, if the local and external case-bases are merged, the two external cases (E1 and E2) will be most similar to the target problem. E2, the closest case, will be used to solve the target, resulting in a comparatively poor solution compared to the local case L1. However, if cases in the external case-base systematically produce worse solutions for a given similarity level, MCBR can take that into account. By learning a sufficiently high dispatching threshold (thereby preferring to solve problems using local cases), MCBR can use only those external cases that are so much more similar that their similarity counterbalances their lower expected solution quality for a given similarity level. In the figure, the dotted line on the x-axis indicates a possible dispatching threshold to avoid the previous problem. In MCBR, because the distance from L1 to the target problem is less than the dispatching threshold, L1's (higher quality) solution is used. To illustrate using the real estate appraisal domain, a realtor may generate a better estimate pricing a new house based on a somewhat dissimilar house that is nearby, instead of a more similar house in a different area. However, using cases from another area might be worthwhile if local cases are too dissimilar (e.g., pricing an apartment when the only local cases are for houses).

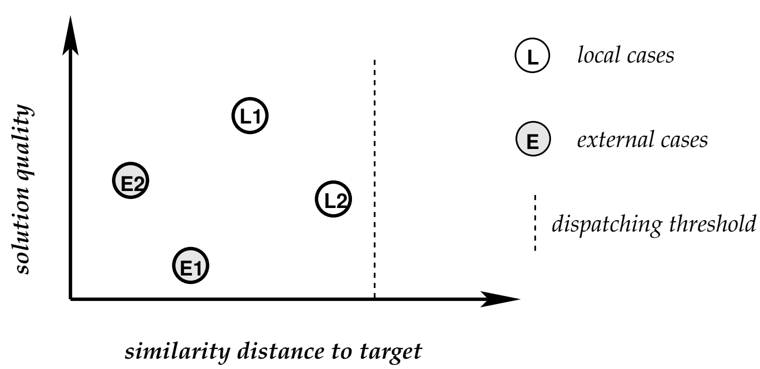


Figure 3: Why case-base merging may decrease accuracy.

Thus we hypothesize that with the right dispatching criteria, MCBR can improve solution quality compared to eager merging, by taking into account the quality of the external case-base and cross-case-base adaptation to select cases that are less similar, but are expected to provide better solutions. Eager merging lacks that capability.

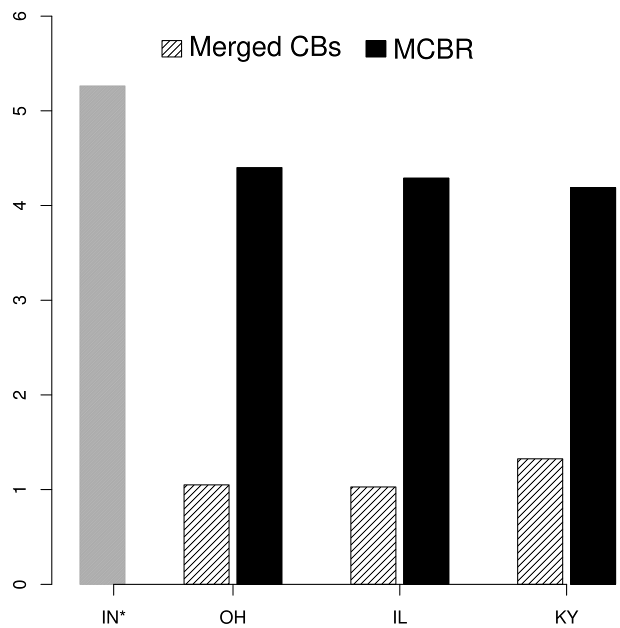


Figure 4: Average k-NN distance for problem-solving with the initial case-base, merged case-bases, and MCBR.

7.3. *Experimental Results on MCBR Managing the Similarity-Quality Tradeoff*

To study this hypothesis, we performed an experiment to determine whether, on average, MCBR was in fact drawing on less similar cases. The experimental design is similar to that of the earlier section, which showed that MCBR produced improved performance compared to eager merging. In this experiment, however, the average nearest neighbor distance (NN-distance) between target problems and selected solution cases was measured for each condition. For both the sparse case-base configuration and the merged case-base configurations, the average NN-distance for each case in the test suite during cross validation was recorded, and these average NN-distances were then averaged across all cases in the test suite.

For the MCBR configuration, for each case in the test suite, the average distance of neighbors used for k-Nearest-Neighbor (k-NN) distance was measured for the cases used to solve the problem, based on whether the case was handled locally or externally. These results are plotted in Figure-4. The average nearest neighbor distance in the initial sparse case-base is 5.26. In the three merged conditions, the average k-NN distance was approximately 1; in the MCBR condition the average k-NN distance was approximately 4. Thus as expected, the average distance decreases as case-bases are merged, and decreases markedly compared to both the initial case-base and the MCBR condition. However, as Figure 2 showed, MCBR performance was superior to that with both the initial case-base and the merged case-bases. Thus in the merged configuration, the cases imported from the external case-base are being deemed more similar and are selected for problem-solving, but are impairing performance. This provides support for the hypothesis that MCBR assists in managing the tradeoff between similarity and usefulness for generating high-quality solutions.

8. MCBR to Guide Use of Case-Bases with Differing Quality

If the explanation in the previous section is correct, MCBR should be beneficial when there are differences between the expected quality of solutions from external and local case-bases. To examine this hypothesis, we conducted an experiment in which we controlled the difference in expected quality by adding noise to the external case-base. This experiment also tested the ability of our learning strategy for case dispatching to automatically adjust to changing case-base characteristics.

In this experiment, we took two random samples of the IN case-base, treating one as the local and the other as the external case-base. To study the effects of case-base quality, varying amounts of uniformly-distributed noise were probabilistically added to the solution part of the cases in the external case-base. Our previous experiment on case-base merging vs. MCBR was repeated on this combination of local and external case-bases. Figure 5 graphs our results. We see that when a noisy case-base is merged, performance of the combined case-base drops. However, the dispatching learning strategy associated with MCBR is able to respond to the noisiness of the external case-base and its reduced contribution to forming an effective solution. Hence as noise increases in the external case-base, the dispatching learning strategy effectively shuts down case-dispatch, handles input problems locally, and thereby avoids the performance degradation that results when the case-bases are merged. This shows that when an external case-base is noisy, MCBR can automatically self-adjust the whole system to compensate for decreased effectiveness of the external case-base, again providing accuracy benefits compared to eager merging. This is consistent with our prediction.

9. Using MCBR to Guide Effective Case Addition

Although merging an entire external case-base may be detrimental when solutions in the external case-base are unreliable for local tasks, it is clear that case additions may be beneficial, even from a case-base that is less reliable than the local one. For example, a new case from an external case-base may fill a crucial competence gap. We hypothesized that, given an original case-base and another case-base

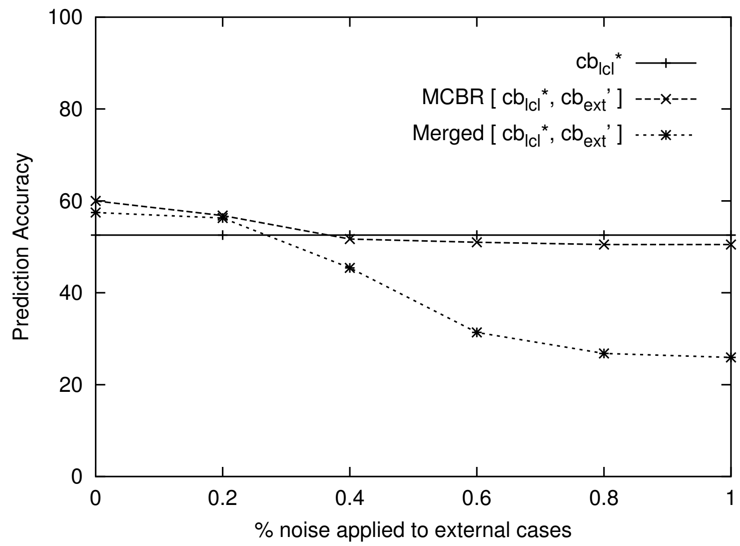


Figure 5: Comparative performance of case acquisition strategies: CB_{lcl} denotes the local case-base (5% IN) and $\text{CB}_{ext'}$ denotes the external noisy case-base (10% IN with added noise).

with less reliable solutions, MCBR is a good strategy for selecting which cases to import from the external case-base. To explore this hypothesis, we performed an experiment in which a very sparse (1%) version of the Indiana case-base (IN*) was the local case-base, and FL was the external case-base. The test simulated conditions in the early phases of using a CBR system, when system competence is low and case addition is needed to increase competence.

We compared the prediction accuracy obtained by (1) MCBR-guided case addition, using MCBR to determine when to import cases from the FL case-base and adding the cross-case-base adapted versions of those cases to the local case-base, and (2) random additions, selecting cases from the FL case-base to add to the local case-base after cross-case-base adaptation. In both conditions, all tests added new cases in blocks of 10, and the best cross-case-base adaptation strategy was automatically selected at the start of the tests and applied to all added cases.

The performance was measured as the percentage of problems successfully solved. The results for MCBR-guided case addition are graphed in Figure 6, along with three base-line comparisons: performance using IN* alone, performance using MCBR to solve problems, but without importing the cross-adapted cases, and performance after merging IN* with cross-case-base adapted versions of all the cases in the FL case-base. Note that in the figure, the line for the average performance of IN* alone overlaps with the performance of MCBR with IN* and FL. Because the characteristics of IN and FL are very different, the capacity of MCBR to draw on the FL

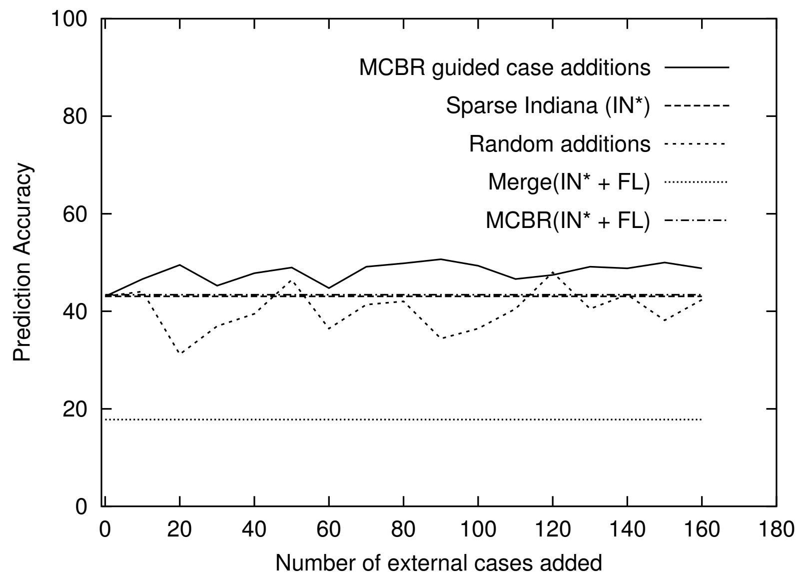


Figure 6: Performance during case-base growth for IN* as local case-base and FL as external case-base.

case-base does not benefit performance. However, nor does it harm performance; the MCBR dispatching strategy is effective at preventing erroneous use of FL cases. In contrast, merging IN* and FL produces markedly worse performance. The same set of experiments was repeated using OH as the external case-base and the results are given in Figure 7. Both of these figures represent the performance of a single instance of IN*.

In both configurations, randomly adding cross-adapted cases from the external case-base results in erratic performance changes, sometimes (as in Figure 6), resulting in worse performance than with IN* alone. However, adding only those cases selected during the MCBR process provides more consistent performance improvement, always outperforming IN* and generally outperforming the other methods. The results suggest that MCBR may provide useful guidance for case addition during initial system use.

It is interesting to note that the performance of the case-base with additions chosen by MCBR exceeds the performance achieved by simply relying on MCBR to draw on those same external cases as needed. We have observed this behavior across various configurations, and plan to investigate it further.

10. Future Directions

As described previously, inter-case-base differences may have a strong effect on the expected usefulness of cases with given similarity levels. Although our experi-

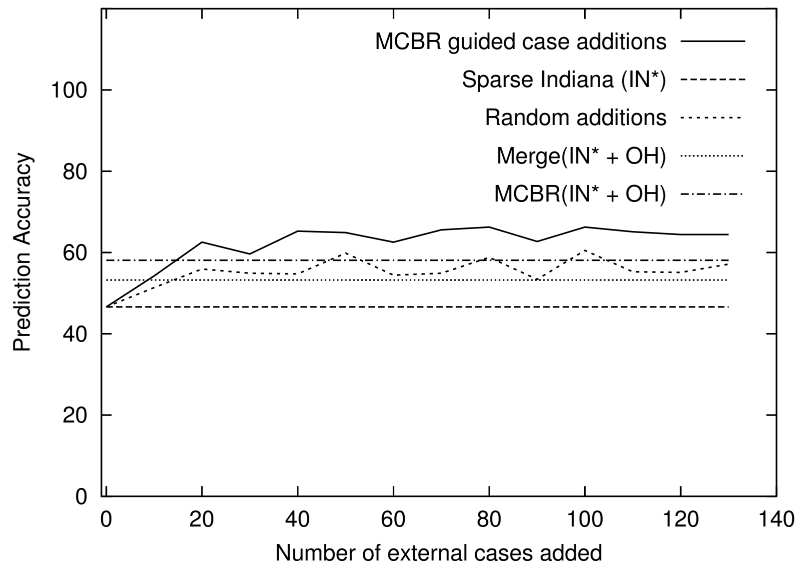


Figure 7: Performance during case-base growth for IN* as local case-base and OH as external case-base.

ments show that threshold-based dispatching can help address this when local and external case-bases have comparable similarity metrics, we anticipate that better performance could be achieved by a richer model of the relationships between the similarity criteria of the local and external case-bases. For example, our current studies use the same similarity criteria to guide retrievals in all case bases, but similarity criteria could be significantly different in different case-bases. In such circumstances, it may be necessary to augment cross-case-base case adaptation with cross-case-base query adaptation, to adjust indices for external case-bases. This adjustment of similarity criteria might help alleviate some of the problems arising from eager merging, and is an interesting area for future study. Likewise, an interesting question is how best to address differences in case representation.

To date, our research focuses on "knowledge light" strategies for MCBR. How additional knowledge could be used within the MCBR process, how to decide when to use MCBR, rather than merging, and how to refine the dispatching threshold over time are all interesting open questions. Because our learning methods depend on analysis of performance for a subset of the problem stream, studies on mining data streams[22] and on estimating the contents of data sets with small samples[23] are also relevant.

Our tests suggest that MCBR may also have value for selective case addition. This potentially includes updating legacy case-bases, by treating any already-updated cases as the "local" case-base and the legacy case-base as the "external" case-base, and automatically importing and cross-case-base adapting the legacy cases needed during processing.

11. Conclusion

Multi-case-base reasoning (MCBR) enables an agent to selectively augment its own case-base as needed, drawing on external case-bases and adjusting their solutions for inter-case-base differences. MCBR is often motivated by the desire to support experience-sharing without requiring access to complete external case-bases and without requiring local storage for all cases. Although the advantages of MCBR are clear when those motivations are significant, this appeared to limit the usefulness of MCBR. This article shows that MCBR may have two additional benefits: Increasing solution accuracy compared to eager merging, and guiding case-base building. For a given local case-base and external case-base for a similar but different task, MCBR can improve accuracy compared to merging both case-bases into a single case-base, even if the same cross-case-base adaptation process is applied to both cases-bases. The article hypothesizes an explanation of this behavior in terms of a tradeoff between similarity of problems and similarity of solution contexts, and demonstrates that the expected benefits also apply when using local and external case-bases developed for the same task environment, but with differing levels of reliability. These results make a case for the broader usefulness of MCBR.

12. Acknowledgments

This research is supported in part by NASA under award No NCC 2-1216 and by the National Science Foundation under award EIA-0202048. This article is revised and expanded from Leake and Sooriamurthi (2003).[24] We thank the anonymous reviewers for their very helpful comments, and thank Hun Myoung Park for statistical consulting.

13. References

[1] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–52, 1994.
[2] J. Kolodner. *Case-Based Reasoning.* Morgan Kaufmann, San Mateo, CA, 1993.
[3] D. Leake, editor. *Case-Based Reasoning: Experiences, Lessons, and Future Directions.* AAAI Press/MIT Press, Menlo Park, CA, 1996.
[4] C. Riesbeck and R.C. Schank. *Inside Case-Based Reasoning.* Lawrence Erlbaum, Hillsdale, NJ, 1989.
[5] D. Leake and R. Sooriamurthi. When two case bases are better than one: Exploiting multiple case bases. In *Case-Based Reasoning Research and Development: Proceedings of the Fourth International Conference on Case-Based Reasoning, ICCBR-01*, pages 321–335, Berlin, 2001. Springer-Verlag.
[6] D. Leake and R. Sooriamurthi. Managing multiple case-bases: Dimensions and issues. In *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference*, pages 106–110, Menlo Park, 2002. AAAI Press.

[7] D. Leake and R. Sooriamurthi. Automatically selecting strategies for multi-case-base reasoning. In S. Craw and A. Preece, editors, *Advances in Case-Based Reasoning: Proceedings of the Fifth European Conference on Case-Based Reasoning*, pages 204–219, Berlin, 2002. Springer Verlag.

[8] P. Cunningham, N. Nowlan, S.J. Delany, and M. Haahr. A case-based approach to spam filtering that can track concept drift. Technical Report TCD-CS-2003-16, Computer Science Department, Trinity College Dublin, 2003.

[9] D. Aha and K. Branting. Stratified case-based reasoning: Reusing hierarchical problem solving episodes. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 384–390, San Francisco, August 1995. Morgan Kaufmann.

[10] B. Smyth and P. Cunningham. A hierarchical case-based reasoning - integrating case-based and decompositional problem-solving. *IEEE Transactions on Knowledge and Data Engineering*, 13(5), 2001.

[11] L. McGinty and B. Smyth. Collaborative case-based reasoning: Applications in personalised route planning. In *Case-Based Reasoning Research and Development: Proceedings of the Fourth International Conference on Case-Based Reasoning*, Berlin, 2001. Springer Verlag.

[12] S. Ontañón and E. Plaza. Learning when to collaborate among learning agents. In *Machine Learning: ECML 2001*, pages 395–405, Berlin, 2001. Springer-Verlag.

[13] S. Ontañón and E. Plaza. Collaborative case retention strategies for cbr agents. In *Case-Based Reasoning Research and Development: Proceedings of the Fifth International Conference on Case-Based Reasoning, ICCBR-03*, Berlin, 2003. Springer-Verlag.

[14] P. Cunningham and G. Zenobi. Case representation issues for case-based reasoning from ensemble research. In *Case-Based Reasoning Research and Development: Proceedings of the Fourth International Conference on Case-Based Reasoning*, Berlin, 2001. Springer Verlag.

[15] Kurt D. Bollacker and Joydeep Ghosh. A supra-classifier architecture for scalable knowledge reuse. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 64–72. Morgan Kaufmann, 1998.

[16] D. Leake, B. Smyth, Q. Yang, and D. Wilson, editors. *Maintaining Case-Based Reasoning Systems*. Blackwell, 2001. Special issue of *Computational Intelligence*, 17(2), 2001.

[17] B. Smyth and E. McKenna. Building compact competent case-bases. In *Proceedings of the Third International Conference on Case-Based Reasoning*, pages 329–342, Berlin, 1999. Springer Verlag.

[18] E. McKenna and B. Smyth. Competence-guided case discovery. In *Research and Development in Intelligent Systems XVIII: Proceedings of ES 2001*, Berlin, 2001. Springer-Verlag.

[19] 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.

[20] AnHai Doan, Alon Halevy, and Natasha Noy, editors. *Proceedings of the Semantic Integration Workshop*, volume 82. CEUR, 2003. http://SunSITE.Informatik.RWTH-Aachen.de/Publications/CEUR-WS/Vol-82/.

[21] D. McSherry. Automating case selection in the construction of a case library. *Knowledge-Based Systems*, 13(2-3):133–140, 2000.

[22] P. Domingos and G. Hulten. Mining high-speed data streams. In *Knowledge Discovery and Data Mining*, pages 71–80, 2000.

[23] J. Callan. Query-based sampling of text databases. *ACM Transactions on Information Systems*, 19(2):97–130, 2001.

[24] D. Leake and R. Sooriamurthi. Dispatching cases versus merging case-bases: When MCBR matters. In *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference*, pages 129–133, Menlo Park, 2003. AAAI Press.