Case-based Plan Diversity

Alexandra Coman, Héctor Muñoz-Avila

Department of Computer Science and Engineering, 19 Memorial Drive West,
Lehigh University, Bethlehem, PA 18015
{alc308, hem4}@lehigh.edu

Abstract. The concept of diversity was successfully introduced for recommender-systems. By displaying results that are not only similar to a target problem but also diverse among themselves, recommender systems have been shown to provide more effective guidance to the user. We believe that similar benefits can be obtained in case-based planning, provided that diversity-enhancement techniques can be adapted appropriately. Our claim is that diversity is truly useful when it refers not only to the initial and goal states of a plan, but also to the sequence of actions the plan consists of. To formalize this characteristic and support our claim, we define the metric of "plan diversity" and put it to test using plans for a real-time strategy game, a domain chosen for the simplicity and clarity of its tasks and the quantifiable results it generates.

Keywords: diversity, similarity, case-based planning

1 Introduction

"Diversity", the quantifiable variation among retrieved query results, has been explored as a means of improving the performance of recommender systems [1,2,3,4,5]. Results that not only are similar to a user query or adhere to a set of constraints, but are also diverse among themselves are argued to provide genuine and useful alternatives, covering a larger portion of the solution space [1].

We believe that the introduction of diversity considerations in case-based planning [6,7,8,9], while so far insufficiently explored, can prove equally advantageous and have a significant impact on fields such as interactive planning [10,11]. In interactive planning, the user is presented with a set of planning choices. With the assistance of the system, the user chooses one that best accomplishes their goals. Plan diversity could enhance such systems by providing the user with truly distinct choices.

In identifying plan matches, one could supplement the criterion of similarity to a new problem-case by that of diversity between selected plans, with benefits similar to those obtained with recommender systems. Hereinafter, we explore the relative merits of several alternatives that can be taken in integrating diversity in plan retrieval. We formally define and assess two types of diversity, each targeting a different aspect of analyzed and retrieved cases in case-based planning.

- **State diversity** - The "weaker" of the two, as we aim to prove, introduced for comparison purposes. It characterizes cases that are dissimilar in terms of

initial and, possibly, final state, but may well be made up of the same (or very similar) plans.

- **Plan diversity** allows the identification of sets of plans with considerably different sequences of intermediary actions, which we hypothesize represent genuine alternatives. Our claim, based on immediate intuition and put to test experimentally, is that diversity is truly useful when it refers not only to the initial and goal states, but rather to the sequence of intermediary steps between them, and is balanced with initial and goal state similarity.

We incorporate both of these forms of diversity into several retrieval methods: two devised specifically for plan retrieval, so as to obtain diversity without sacrificing similarity; and a plan-diversity-aware adaptation of "Bounded Greedy", a technique previously introduced for similar purposes in recommender systems [1,2,3]. For comparison purposes, we also test the standard "Bounded Greedy" algorithm, which is based on what we refer to as "state diversity". The preservation of similarity to the query, while promoting diversity between matches, is a constraint we inherit from previous forays into diversity [2] and which we attempt to address in a manner suitable to the plan retrieval domain.

We test all aforementioned methods on a series of cases containing the usual components (initial state, goal state, plan) of cases in case-based planning [7]. Our testbed is Wargus, a real-time strategy game, chosen because it has been used in previous work (e.g., [8]) and for the possibilities it offers to obtain easily quantifiable results. Diversity, or the lack of it, is immediately observable on watching a game unfold, but also relevantly reflected, for the purpose of analysis, in quantifiers such as game scores and the duration of a game session. Our goal to achieve diversity for case-based planning is motivated by the success obtained with recommender systems, which thus far has been mostly demonstrated for analysis tasks. For our case-based plan diversity methods, the results are very encouraging: adapted plans retrieved based on our case-based plan diversity methods generate game-play instances of significant and discernible variation.

It should be noted that we are, at this stage, only marginally concerned with how "successful" these diverse retrieved plans will, on average, be in solving whatever problem they are meant to solve within any given parameters specific to their domain. By "success", we will, instead, refer to generating a set of adapted plans that produce results running the gamut from low to excellent over a variety of criteria. We consider plan variation to be intrinsically valuable and a goal in itself, although the exact nature of its value, as well as the range of the possibilities it opens up, is bound to vary from domain to domain.

2 Background

McGinty and Smyth [1] enhance the typical recommendation cycle to include the notion of diversity, using a technique called "bounded greedy", which was first introduced in [2]. "Bounded greedy" works by first ranking cases based on their similarity to the query and afterwards repeatedly selecting, out of the ranked list,

those that maximize the weighted sum of similarity to the query and "relative diversity", where relative diversity is defined as (*C* is a set of cases, *n* the number of cases in *C*, *c* a case and *Sim* a similarity metric):

$$RelDiv(c, C) = \frac{\sum_{c_i \in C}(1 - Sim(c_i, c))}{n}, C \neq \{\}. \quad (1)$$

The selected cases are the ones which maximize the quality given by:

$$\alpha Sim(q, i) + (1 - \alpha) RelDiv(i, R) \ , \quad (2)$$

where q is the user query and R the set of cases retrieved so far. Their methods are shown to be an improvement over classical similarity-based retrieval.

While, admittedly, various aspects of the recommendation cycle are not relevant to case-based planning, we can easily retain the diversity metric and adjust it to meet our own purposes. An important difference between recommender systems, and analysis tasks in general, and synthesis tasks such as case-based planning is, however, bound to affect the proper handling of such an adjustment: a classification task stops with the identification of a satisfactory query result or set of results, whereas case-based planning (as synthesis task) must, after identifying a query result, adapt it to produce a solution plan, which, in our context, must be executed.

How do we describe one such satisfactory set of plans with regard to diversity? A successful diversity-aware retrieval algorithm is one that generates plans that, when adapted, produce diverse results.

We follow the usual case-based planning convention: cases are represented as having two components: the problem, composed of initial and goal states, and a plan transforming the initial state into the goal state [7].

3 Example

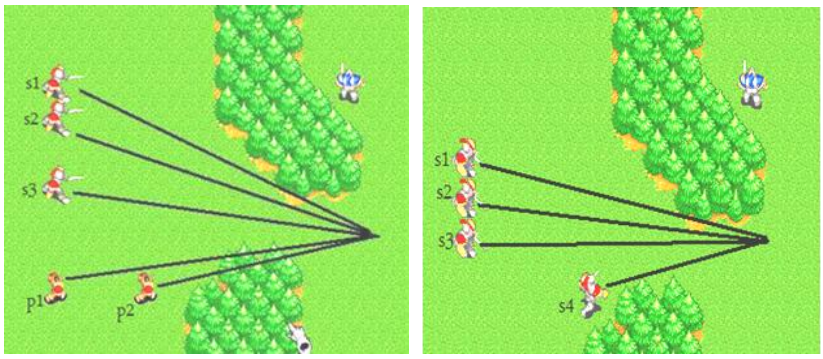
We use a real-time-strategy game to showcase an example illustrating why diversity based on initial and final states ("state diversity") is insufficient to produce enough variation in the retrieved cases. Approaches to game-play in real-time-strategy games (and, generally, any game genres based on simulating combat) can be categorized by strategies consisting of some combination of "offensive" and "defensive" measures. Any such strategy, however complex, is bound to be reducible to combinations of these two basic approaches, in varying forms and degrees. Furthermore, most actions that can be taken in a game, such as using various types of attacks, building a defensive unit, "healing" one's units or fleeing can usually be categorized as pertaining more to a defensive or to an offensive strategy. Ideally, the system would retrieve plans that are diverse as per these categories. The problem is that a significant effort would be required to annotate the plans according to these categories. Instead, we propose to use plan diversity as the means to identify diverse plans without requiring any such plan labeling to be known beforehand.

Our experiments (described in more detail in Section 5) involve retrieving (based on diversity considerations, as well as similarity to a "new problem" case)

gameplay plans, wherein a plan consists of a series of actions (offensive, defensive or balanced) that the player's units can take. Its initial state is the initial structure of our player's combat force i.e. number of units of each kind: in our experiments, these can be "peasants" or "soldiers". We present an example of how plan retrieval would be handled in this context, when using state and plan diversity. Let the plan library contain only the following three cases and let us, for the moment, assume that we are only looking for a pair of maximally diverse cases (ignoring the criterion of similarity to a new problem):

Case 1: The initial state configuration consists of three "soldier" units and two "peasant" units. The plan consists of all units attacking.

Case 2: The initial state configuration consists of four "soldier" units and no "peasant" units. The plan, once again, consists of all units attacking.



(C1) **(C2)**

(C3)

Fig. 1. Case pairs (C1, C2) and (C2, C3) are state-diverse, while pairs (C1, C3) and (C2, C3) are plan-diverse. C1 and C2 are instances of the same offensive strategy (all units attack), while C3 exemplifies a distinct defensive strategy ("soldier" units patrol, "peasant" units stay put).

Case 3: The initial state configuration consists of three "soldier" units and two "peasant" units (it is identical to that of Case 1). The plan consists of the "soldiers"

patrolling given areas in expectation of an attack, while the weaker "peasant" units stay put, not "willingly" exposing themselves to damage.

Cases 1 and 2 display purely offensive strategies, whereas Case 3 is largely defensive. An algorithm based on "state diversity" (diversity of initial states only, in our experiment, for reasons explained in Section 5) would select either Case 1 and Case 2 or Case 2 and Case 3 (either Case 1 or Case 3 would always be discarded, as they have identical initial states).

Assuming, therefore, that "tie" situations such as this are handled by choosing one of the multiple cases with identical initial states randomly, the probability of choosing truly diverse plans tends to decrease with the total number of cases. In this simple example, we have a 0.5 probability of obtaining two offensive strategies, which, on being adapted for a new problem case, translate to identical strategies (it is only reasonable that an adaptation of an offensive plan will also be offensive).

If retrieving based on plan diversity, however, we are guaranteed to obtain either Case 1 and Case 3 or Case 2 and Case 3. That is, with a probability of 1, we will (in this simple example) be presented with two plans which are truly different in the strategy that they incorporate and will generate distinct adaptations.

4 State and Plan Diversity

Herein, we describe the diversity-aware plan-retrieval methods we propose and evaluate. For all algorithms below, we assume cases consisting of triples of the type (*initial state*, *goal state*, *plan*) and a new problem described only in terms of the initial and the goal state. The problem of finding a set of cases that are maximally diverse from one another and, at the same time, maximally similar to the problem is computationally expensive [2]. As a result, the algorithms below aim, instead, at finding good approximations of such optimal solutions.

4.1 State Diversity through Similarity Clusters

Below, we show the cluster-based retrieval method for state diversity, which we call *SDSC - State Diversity through Similarity Clusters*. First, cases are sorted in reverse order of their similarity to the query problem (line 1). Cases that are similar to one another are clustered together. To obtain k cases that are likely maximally similar to the new problem, as well as diverse from each other (such that there are no two identical similarity scores in the retrieved set) we need only choose one case from each of the first k clusters, as "state diversity" is also based on the initial and final states (line 4). When we choose a case from a cluster, we do so randomly.

Procedure SDSC(newProb, CB)
Input: newProb: the query problem; CB: the case base
Output: R: the list of recommended cases in CB for newProb
1. CB' ← sort CB in decreasing order of their similarity to newProb
2. nC ← number of clusters in CB'
3. R←{}
4. **for** j←1 **to** nC **do**
   simClust ← select-cluster(j, CB')
   case ← select-random-case(simClust)
   R ← R ∪ {case}
   end-for
5. **return** R

4.2 Plan Diversity through Greedy Selection

Below, we show the retrieval algorithm based on plan-diversity, which we call *PDGS - Plan Diversity through Greedy Selection*. Cases are sorted in reverse order, based on their similarity to the new problem (line 1). We add to R the case in CB' which is the closest to the target problem (line 3). For each case i, starting with the second-highest ranking in the hierarchy, we compute plan diversity (*plDiv*) between it and the cases chosen so far (Formula 3 below). The domain-specific similarity metric *plSim* identifies two plans as similar based on their sequences of actions. If *plDiv* is higher than threshold Δ and *stSim* (similarity to the new problem) is higher than threshold Δ', then c is added to the hierarchy. Otherwise, stop and return the chosen case set (lines 4-6).

$$plDiv(case_i, chosenCaseSet) = \sum_{k=1,|chosenCaseSet|} \frac{1-plSim(case_i,case_k)}{|chosenCaseSet|}. \quad (3)$$

4.3 Plan-Diversity Bounded Greedy

Below, we show our adapted version of the "bounded greedy" algorithm [1], which we call *Plan-Diversity Bounded Greedy - PBGA*. Our version works by selecting, on each step, the case that maximizes the sum of the similarity to the new problem and plan diversity with regard to the states selected so far, according to the following formula (which is a variant of Formula 2):

$$simPlDiv = \alpha * Sim(newProb, c) + (1 - \alpha) * plDiv(c, R)\,, \quad (4)$$

where *plDiv* is plan diversity as described in Formula (3). For the original version of "Bounded Greedy", we compute state diversity according to the formula (*simSt* is a similarity metric comparing the initial and goal states):

$$stateDiv(case_i, chosenCaseSet) = \sum_{k=1,|chosenCaseSet|} \frac{1-simSt(case,case_k)}{|chosenCaseSet|}. \quad (5)$$

The original "Bounded Greedy" algorithm [1] differs from our variant below in that they use state similarity and diversity to select cases to be added to R (lines 2-3), whereas "Plan-Diversity Bounded Greedy" uses state similarity and plan diversity.

Procedure PDGS(newProb, CB, Δ, Δ')
Input: newProb: the query problem; CB: the case base; Δ, Δ': thresholds
Output: R: the list of recommended cases in CB for newProb
1. CB' ← sort CB in decreasing order of their similarity to newProb
2. R←{}, i←2
3. **add** first case in CB' **into** R
4. **repeat**
   c ←select case i from CB'
   if plDiv(c, R) > Δ **and** stSim(newProb,c) > Δ' **then**
      R ← R ∪ {c}
   end-if
   i←i+1
5. **while** (plDiv(c, R) > Δ **and** stSim(newProb,c) > Δ' **and** i≤$|CB|$)
6. **return** R

Procedure PBGA(newProb, CB, k)
Input: newProb: the query problem; CB: the case base; k: integer
Output: R: the list of k recommended cases in CB for newProb
1. R ←{}
2. **for** i←1 **to** k **do**
   Sort CB by simPlDiv
   c ←first case in CB
   R←R ∪ {c}
   CB ←CB – {c}
3. **end-for**
4. **return** R

5 Experiment

As with other CBR research, we use a game as our testbed [8,12]. Our hypothesis is that plan retrieval by taking into account plan-diversity considerations will result in a wider range of choices than state-diversity-based retrieval. On adapting plan-diverse retrieved cases and running these in a game, we expected to obtain results (measured via game-specific metrics) that are quantifiably more varied than those obtained by running plans retrieved via state-diversity-based methods.

5.1 Experimental Setup

Our experiments are conducted using simple real-time-strategy game plans, run on "Wargus", a clone of "Warcraft II: Tides of Darkness" which uses the free real-time strategy game engine "Stratagus".

The two-player Wargus games we stage take place on a 32x32 tile map, with our player acting out plans against the built-in Wargus enemy AI. We allow only two types of units: "soldiers" (basic fighting units) and "peasants" (used normally for resource harvesting and creating "building" units, but introduced here for the purpose of illustrating strategy variation by having weaker units engage in battle). We only vary our player's initial configuration. State similarity and, subsequently, state diversity, are based on the initial configuration (our player's units). Plans are not annotated by the specific goals they achieve (e.g., capture the center of the map). This simulates a situation in which successful plans are captured by observing the user's actions, but not knowing their intent. Thus, an initial state is defined by a pair of type *(numSold, numPeas)*, where *numSold* is the number of "soldier" units and *numPeas* the number of "peasant" units our player has at their disposal.

The formula for state similarity, to be utilized in diversity-aware retrieval algorithms as described in the previous section, therefore becomes:

$$simSt(case_1, case_2) = \frac{\frac{min(numSold_1, numSold_2)}{max(numSold_1, numSold_2)} + \frac{min(numPeas_1, numPeas_2)}{max(numPeas_1, numPeas_2)}}{2}. \quad (6)$$

As for the plans themselves, the actions they can include are "AttackMove" (moving to a given spot on the map, while attacking any enemy units encountered on the way), "Patrol" (a defensive attitude, consisting of moving between two locations, prepared to fight in case of an enemy invasion attempt), "Move" (moving the unit to specified coordinates) and the self-explanatory "Attack". For the sake of simplicity, we do not take into account any coordinates associated with these moves when computing plan similarity. Groups of units will, as a rule, move in the same direction, both in our initial "library" plans and in our adapted ones. We, also, do not consider action order for computing plan similarity, as it is neither relevant to this particular task, nor vital in demonstrating the basic concept of plan diversity. Plan similarity between two plans is defined by:

$$plSim(case_1, case_2) = \frac{\frac{min(numAttack_1, numAttack_2)}{max(numAttack_1, numAttack_2)} + \frac{min(numMoveAttack_1, numMoveAttack_2)}{max(numMoveAttack_1, numMoveAttack_2)}}{4} + \frac{\frac{min(numPatrol_1, numPatrol_2)}{max(numPatrol_1, numPatrol_2)} + \frac{min(numMove_1, numMove_2)}{max(numMove_1, numMove_2)}}{4}. \quad (7)$$

The new problem case will be defined by its initial state, a *(numSoldiers, numPeasants)* pair. We use a new problem with the initial state (10, 9): 10 "soldiers" and 9 "peasants" (see Fig. 2).

Our "plan library" contains 16 plans (Table 1), consisting of all possible state-strategy combinations between 4 start states (with varying numbers of "peasant" and "soldier" units) and a number of plans.

Fig. 2. Initial configurations of a "library" case and a "new problem". The layout of the terrain and lack of building units ensure the brevity of game-play episodes, making them easily comparable to each other.

Table 1 shows only a summarization of the plans; the following is an example of an actual plan, as stored in the plan library:

```
m_pb.attackMove(1,13,10);
m_pb.move(2, 7, 8);
m_pb.patrol(2, 9, 7);
m_pb.move(13, 5, 9);
m_pb.patrol(13, 6, 10);
```

It instructs unit 1 to move to coordinates (13,10) on the map, while attacking any enemy unit encountered on the way; and units 2 and 13 to move to coordinates (7,8) and (5,9), respectively, and to start patrolling back and forth between their new location and coordinates (9,7) and (6,10), respectively.

Although the following information is not stored in the plan library, conceptually, there are 4 plan strategies (in each case, adapted to the number of units in the start state). The four strategies are: (1) "Offensive" (all units attack), (2) "Defensive" (all "soldier" units patrol and all "peasant" units stay put), (3) "Balanced Offensive" (75% of "soldiers" attack, 25% patrol), and (4) "Balanced Defensive" (50% of "soldiers" attack, 50% patrol).

Adaptation is performed by building a plan based on the same strategy as the retrieved plan, but adjusted to the number of units in the new problem initial state. For example, if the retrieved plan is Case 2 in Table 1, a "defensive" plan, the adapted plan for the new problem will have all 10 "soldier" units move to a key location and patrol, while all 9 "peasant" units remain where they are.

Our experiments are conducted as follows:

- Each of the four retrieval algorithms ("State Diversity through Similarity Clusters", "Plan Diversity through Greedy Selection", "State Diversity Bounded Greedy" and "Plan Diversity Bounded Greedy") is run on the new problem and set of library cases 4 times (tie-breaking is handled by randomly selecting a case), each time recording the top 4 retrieved plans. For *PDGS* (see Section 4.1.2), we use the thresholds $\Delta = 0.3$ and $\Delta' = 0.5$. For both "Bounded Greedy" variants, α is set at 0.5.

- Retrieved plans (the top 4) are adapted to the new problem and the resulting sequences of actions are run in the game. At the end of each such game (after all enemy units have been destroyed), the values of two metrics are recorded: *number of game cycles* (as recorded by Wargus) and *score* (consisting of the difference between the player's score and the opponent's score, as computed by Wargus. A player's score is incremented when an opponent's unit is destroyed).

Table 1. Each case consists of a state and a plan. The table shows summarizations of the states and of the plans, rather than the actual states and plans stored, for the sake of space.

CASES		
	Initial state	**Plan summarization**
1.	8 s, 3 p	AttackMove x 11
2.	8 s, 3 p	Move x 8, Patrol x 8
3.	8 s, 3 p	AttackMove x 6, Move x 2, Patrol x 2
4.	8 s, 3 p	AttackMove x 4, Move x 4, Patrol x 4
5.	3 s, 2 p	AttackMove x 5
6.	3 s, 2 p	Move x 3, Patrol x 3
7.	3 s, 2 p	AttackMove x 2, Move x 1, Patrol x 1
8.	3 s, 2 p	AttackMove x 1, Move x 2, Patrol x 2
9.	4 s, 0 p	AttackMove x 4
10.	4 s, 0 p	Move x 4, Patrol x 4
11.	4 s, 0 p	AttackMove x 3, Move x 1, Patrol x 1
12.	4 s, 0 p	AttackMove x 2, Move x 2, Patrol x 2
13.	5 s, 5 p	AttackMove x 10
14.	5 s, 5 p	Move x 5, Patrol x 5
15.	5 s, 5 p	AttackMove x 3, Move x 2, Patrol x 2
16.	5 s, 5 p	AttackMove x 2, Move x 3, Patrol x 3

5.2 Results

The results are shown in Fig. 3 (the curves show averaged results of multiple game runs: each point in each of the graphs represents the mean of 4 games). Incorporating the plan diversity criterion in the retrieval process leads to the selection of plans which, after being adapted and run in the game environment, generate significantly varied results (both for score and for game cycles). The variation in results for state diverse plans is negligible in comparison and due to the random factor introduced by the tie-breaking mechanism, as well as to the non-deterministic nature of the game (even when running several games with identical initial configuration and strategies, there will be some variation in game duration and final score). Diversity based on

these factors is not satisfactory, as its consistency cannot be guaranteed over multiple runs.

Both tested plan-diversity-aware algorithms, "Plan Diversity through Greedy Selection" and "Plan-Diversity Bounded Greedy" (with retrieval sets of size 4) always retrieve sets of plans containing all four types of strategies, whereas, with state diversity, the retrieval of a highly diverse set is highly unlikely and, if occurring, largely due to chance.

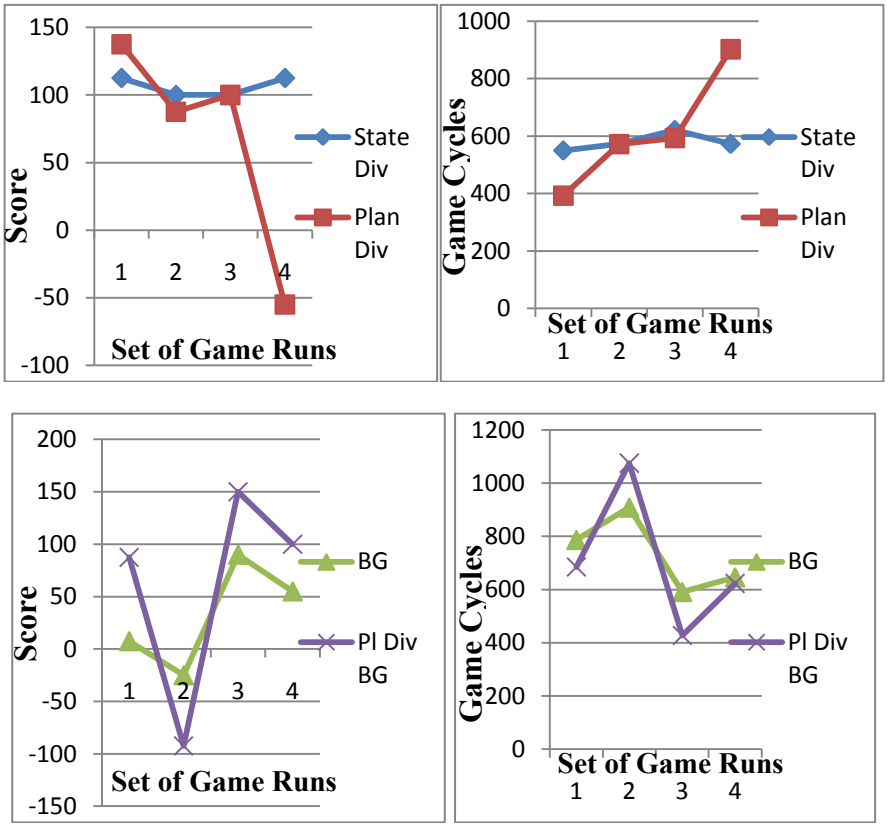


Fig. 3. (a) State Diversity by Similarity Clusters vs. Plan Diversity by Greedy Selection. (b) Bounded Greedy vs. Plan-Diversity Bounded Greedy. Score (left) and game cycles (right) exhibit considerably more variety on adaptations of results based on Plan Diversity. Each point on the x axis represents the average of 4 game runs.

In our State Diversity by Similarity Clusters test runs, at least two results from the state-diverse retrieval set were always instances of the same strategy.

As for the actual results obtained when running these retrieved plans in the game (see Fig. 3), each of the methods based on plan diversity produced results within a larger range than its state-diversity counterpart: the highest and the lowest results, in

terms of score, as well as game cycles, were obtained on running plans retrieved with plan-diversity-aware methods.

For the *PDGS* method, the standard deviation was 211.5 for number of game cycles and 84.3 for score (compare with 29.9 for number of game cycles and 7.2 for score in the case of state diversity). In the *PBGA* case, the standard deviation is approximately twice as large as for regular "Bounded Greedy" (271.4 for number of game cycles and 105.9 for score, compared to 142.9 for number of game cycles and 50.8 for score).

We ran the same experiment on an alternative map (shown in Fig. 4) and obtained similar results. The second map is topologically different from the first one in that there are two gaps in the "forest" initially separating the two opponent "camps", offering more passage possibilities for the "enemy army".

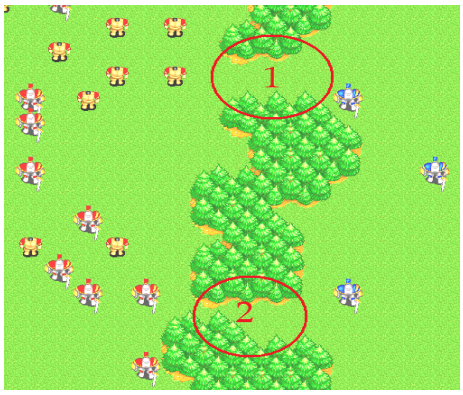


Fig. 4. A second map, topologically-different from that in Fig. 2 (two gaps in the forest between camps), on which we have obtained similar results.

6 Related Work

In previous sections, we referred to the use of diversity with recommender systems and adapted a diversity-enhancing technique previously proposed. Recommender systems (with e-commerce as their most common application) [1,2,3,4,5,13] display, to a particular user, frequently in an individualized manner, sets of items from a solution space that would be difficult to navigate without filtering. Conversational recommender systems [3] work over several cycles: they recommend a set of options, obtain user feedback and repeatedly refine their suggestions based on it, producing new sets of recommendations. Feedback may consist of a simple choice between suggested items or of critiquing [14] of particular features of a suggested item. Diversity can be introduced in the recommendation stage, ensuring that the retrieved kNN-set of most similar cases (which satisfy a series of constraints) [13] is also maximally diverse [1]. Various techniques for achieving this have been proposed: Bridge and Kelly [3], for example, incorporate diversity enhancement in collaborative recommender systems [14], which base recommendations not on features of the items

themselves but on preferences of neighbors (users assessed as being similar to the user to whom recommendations are being made, based on common item ratings), using "Bounded Greedy" with collaborative-data distance metrics. The trade-off between similarity and diversity is an important consideration in choosing a diversity-enhancement technique [2]. McGinty and Smyth [1] use a method called "adaptive selection" to combine similarity and diversity criteria in accordance with the user's feedback on each consecutive set of recommendations for the same query.

As for related work in planning, a framework for summarizing and comparing HTN plans has been proposed [15]. Maximally different plans are identified using a metric called "plan distance", which is conceptually similar to our notion of "diversity" between two plans (in one of its basic forms, "distance" consists of the sum of the number of features which appear in the first plan, but not the second and the number of features that appear in the second plan, but not the first, divided by the total number of features). However, the type of plan differentiation proposed is based on high-level, abstracted, semantically-significant features of plans, rather than their low-level, raw components (states and actions). Furthermore, distance-based comparison is being conducted only within pairs of plans, not within sets of multiple plans. Another related work proposes domain-independent methods for generating diverse plans using distance metrics based on actions, intermediary states and causal links between actions [16]. These reflect similar concerns to those behind our "plan" and "state" diversity. The main difference between all the aforementioned previous work and our own is that the former is knowledge-complete, requiring that complete planning domain knowledge is provided, allowing plan generation from scratch, whereas our work is more in line with the knowledge-light CBR approach; we are adapting retrieved plans, rather than generating new ones from scratch. In fact, in our current Wargus framework, we do not have complete knowledge for plan generation; only the cases, the state and plan similarity metrics, and the adaptation algorithm are known.

The concept of plan distance has also been employed for conducting a comparative evaluation of replanning and plan repair [17]. These two methods adapt to unexpected occurrences during plan execution (by constructing a new plan or adapting the existing one to the new conditions, respectively). A new or adjusted plan produced by either method should be "stable", that is, depart from the original plan only insofar as it is necessary in order to successfully adapt to the new conditions. The smaller the distance between a new plan and the original one, the greater the new plan's stability. Distance is computed as the sum of the number of actions that appear in the first plan, but not the second and the actions that appear in the second plan, but not the first. The work reported in [17] is addressing a problem that seems almost the reverse of our similarity/diversity trade-off. They apply their method in a context in which the difference between plans is desirable only within a very narrow set of parameters (as dictated by the new goals or unexpected execution circumstances) and should, otherwise, be kept to a minimum. We, on the other hand, consider diversity to be intrinsically desirable and explore methods for maximizing it, while also maintaining similarity. In addition, the work of Fox et al. is knowledge-complete, as defined in the previous paragraph.

7 Conclusions

We demonstrate how the concept of diversity, as previously explored in the field of recommender systems, can be successfully adapted to help increase diversity within the sets of retrieved plans in case-based planning. To this end, we have formally defined two diversity metrics ("plan" and "state" diversity) and incorporated them into a series of methods for attaining diverse plan retrieval, which we evaluated comparatively. Our experiments show that methods based on "plan diversity", balanced with initial and goal state similarity, retrieve plans that are discernibly varied in the results they produce once they are executed, therefore representing genuine alternatives.

For future work, we intend to explore how the capability of the plan adaptation algorithm influences the diversity of the resulting cases retrieved; if the adaptation algorithm is very powerful, it is conceivable that it could be used to obtain a variety of solution plans adapted from the retrieved plan. On the other hand, these adapted plans might be too far from the query provided by the user. Hence, we would like to explore how the retrieval-centered mechanism developed in this paper would fare versus an adaptation-centered mechanism such as the one reported in [18].

References

1. McGinty, L., Smyth, B.: On the Role of Diversity in Conversational Recommender Systems. In: K.D. Ashley and D.G. Bridge (Eds.): ICCBR 2003, LNAI 2689, pp. 276–290. Springer, Heidelberg (2003)
2. Smyth B., McClave, P.: Similarity v's Diversity. In: D.W. Aha and I. Watson (Eds.): ICCBR 2001, LNAI 2080, pp. 347-361. Springer, Heidelberg (2001)
3. Bridge, D., Kelly, J.P.: Ways of Computing Diverse Collaborative Recommendations. In: V. Wade, H. Ashman, and B. Smyth (Eds.): AH 2006, LNCS 4018, pp. 41–50. Springer, Heidelberg (2006)
4. McSherry, D.: Increasing Recommendation Diversity Without Loss of Similarity. In: Proceedings of the Sixth UK Workshop on Case-Based Reasoning, pp. 23-31. Cambridge, UK (2001)
5. McSherry, D.: Diversity-Conscious Retrieval. S. Craw and A. Preece, editors. In: T.R. Roth-Berghofer et al. (Eds.): ECCBR 2006, LNAI 4106, pp. 9-29. Springer, Heidelberg (2002)
6. Cox, M.T., Muñoz-Avila, H., Bergmann, R.: Case-based Planning. In: The Knowledge Engineering Review, Vol. 00:0, pp. 1–4. Cambridge University Press, UK (2005)
7. Muñoz-Avila, H., Cox, M.T.: Case-Based Plan Adaptation: An Analysis and Review. In: IEEE Intelligent Systems, pp. 75-81 (2008)
8. Ontañón, S., Mishra, K., Sugandh, N., Ram, A.: On-line Case-Based Planning. In: Computational Intelligence Journal 26(1), pp. 84-119 (2010)
9. Spalazzi, L.: A Survey on Case-Based Planning. In: Artificial Intelligence Review 16, pp. 3–36. Kluwer Academic Publishers, the Netherlands (2001)

10. Cox, M.T., Veloso M.M.: Supporting Combined Human and Machine Planning: An Interface for Planning by Analogical Reasoning. In: D. Leake & E. Plaza (Eds.): Case-Based Reasoning Research and Development: Second International Conference on Case-Based Reasoning, pp. 531-540. Springer, Berlin (1997)
11. Muñoz -Avila, H., McFarlane, D., Aha, D.W., Ballas, J., Breslow, L.A., Nau, D.: Using Guidelines to Constrain Interactive Case-Based HTN Planning. In: Proceedings of the Third International Conference on Case-Based Reasoning, pp. 288-302. Springer, Berlin (1999)
12. Aha, D.W., Molineaux, M., Sukthankar, G.: Case-based Reasoning for Transfer Learning. In: L. McGinty and D.C. Wilson (Eds.): ICCBR 2009, LNAI 5650, pp. 29–44. Springer, Berlin Heidelberg (2009)
13. McSherry, D.: Completeness Criteria for Retrieval in Recommender Systems. In: T.R. Roth-Berghofer et al. (Eds.): ECCBR 2006, LNAI 4106, pp. 9 – 29. Springer, Berlin Heidelberg (2006)
14. Burke, R.: Interactive Critiquing for Catalog Navigation in E-Commerce. In: Artificial Intelligence Review, 18(3-4), pp. 245-267 (2002)
15. Myers, K. L.: Metatheoretic Plan Summarization and Comparison. In: Proceedings of the 16th International Conference on Automated Planning and Scheduling (ICAPS-06), pp. 182-192. AAAI Press (2006)
16. Srivastava, B., Kambhampati, S., T. Nguyen, M. Do, B., Gerevini, A.: Domain independent approaches for finding diverse plans. In: Proceedings of International Joint Conference on AI (IJCAI-07), pp. 2016-2022. AAAI Press (2007)
17. Fox, M., Gerevini, A., Long, D., Serina, I.: Plan Stability: Replanning Versus Plan Repair. In: Proceedings of International Conference on Automated Planning and Scheduling (ICAPS-06), pp. 212-221. AAAI press (2006)
18. Lee-Urban, S., Muñoz-Avila, H.: Adaptation Versus Retrieval Trade-Off Revisited: an Analysis on Boundary Conditions. In: L. McGinty and D.C. Wilson (Eds.): ICCBR 2009, LNAI 5650, pp. 180-194. Springer, Berlin Heidelberg (2009)