Using CBR in the Exploration of Unknown Environments with an Autonomous Agent

Luís Macedo[1,2], Amílcar Cardoso[2]

[1] Department of Informatics and Systems Engineering, Engineering Institute, Coimbra Polytechnic Institute,
3030-199 Coimbra, Portugal
lmacedo@isec.pt
http://www2.isec.pt/~lmacedo
[2] Centre for Informatics and Systems of the University of Coimbra, Department of Informatics, Polo II,
3030 Coimbra, Portugal
{lmacedo, amilcar}@dei.uc.pt

Abstract. Exploration involves selecting and executing sequences of actions so that the knowledge of the environments is acquired. In this paper we address the problem of exploring unknown, dynamic environments populated with both static and non-static entities (objects and agents) by an autonomous agent. The agent has a case-base of entities and another of plans. This case-base of plans is used for a case-based generation of goals and plans for visiting the unknown entities or regions of the environment. The case-base of entities is used for a case-based generation of expectations for missing information in the agent's perception. Both case-bases are continuously updated: the case-base of entities is updated as new entities are perceived or visited, while the case-base of plans is updated as new sequences of actions for visiting entities/regions are executed successfully. We present and discuss the results of an experiment conducted in a simulated environment in order to evaluate the role of the size of the case-base of entities on the performance of exploration.

1 Introduction

Exploration may be defined as the process of selecting and executing actions so that the maximal knowledge of the environment is acquired at the minimum cost (e.g.: minimum time and/or power) [38]. The result is the acquisition of models of the physical environment. There are several applications like planetary exploration [4, 16], rescue, mowing [18], cleaning [12, 36], etc. Strategies that minimize the cost and maximize knowledge acquisition have been pursued (e.g., [2, 3, 10, 22, 25, 35, 38-41]). These strategies have been grouped into two main categories: undirected and directed exploration [38]. Strategies belonging to the former group (e.g., random walk exploration, Boltzman distributed exploration) use no exploration-specific knowledge and ensure exploration by merging randomness into action selection. On the other hand, strategies belonging to the latter group rely heavily on exploration specific-knowledge for guiding the learning process. Most of these directed strategies rely on the maximization of knowledge gain (e.g., [35]). This technique agrees with some psychological studies that have shown that novelty and new stimuli incite exploration in humans (e.g., [6]). Curiosity is the psychological construct that has been closely related with this kind of behavior. However, as argued by Berlyne [6], in addition to novelty, other variables such as change, surprisingness, complexity, uncertainty, incongruity and conflict also determine this kind of behaviour related to exploration and investigation activities. Therefore, in addition to curiosity (or novelty) other motivations such as surprise and hunger seem to influence the exploratory behaviour of humans [19].

Most of these approaches assume that the environment is static. Exceptions are, for instance, the works of [3] and [7]. These works address the problem of acquiring models of the environment where objects change their location frequently.

Most of the environments in which exploration occurs lack a domain theory and are characterized by unpredictability or uncertainty. Therefore, the agent may take advantage of using CBR for dealing with autonomous generation and management of goals as well as plans to accomplish these goals. Besides, together with a Bayesian approach, CBR may be used to deal with uncertainty.

In this paper we describe an approach for the exploration of unknown, dynamic environments populated with static and non-static entities by an agent whose decision-making/reasoning process relies heavily on CBR. The agent is continuously moving in the environment from location to location, visiting unknown entities that inhabit the environment as well as unknown regions. At each time, the agent generates goals that express the intention to visit regions or entities. For each goal a Hierarchical Task Network (HTN) plan [13] is generated. Both the generation of goals and plans is the result of CBR since they are generated from past cases of successful plans. Every time a plan is finished, the case-base of plans is updated with it. Likewise, as exploration is performed, the agent continuously updates its map of the environment with the geometric locations of the entities perceived and updates its episodic memory with those cases of entities built from the entities visited or perceived.

The next section describes how an agent represents entities internally, their geometric locations in the environment (maps) and courses of action (plans). Then, we present the strategy adopted for the exploration of unknown, dynamic environments. The components in which CBR plays a central role, such as the generation of expectations/assumptions and generation of goals and plans, are described. Then, we present and discuss an experiment carried out to evaluate the role of the case-base size in the performance of exploration. Finally, we present conclusions.

2 Agent's Memory

The memory of an agent stores information about the world. This information comprises the configuration of the surrounding world such as the position of the entities (objects and other animated agents) that inhabit it, the description of these entities themselves, descriptions of the sequences of actions (plans) executed by those entities and resulting from their interaction, and, generally, beliefs about the world. This information is stored in several memory components. Thus, there is a metric (grid-based) map [40] to spatially model the surrounding physical environment of the agent. Descriptions of entities (physical structure and function) and plans are stored both in the episodic memory and in the semantic memory [1, 14]. We will now describe in more detail each one of these distinct components.

2.1 Metric Map

In our approach, a (grid-based) metric map (Fig. 1) of the world is a three-dimensional grid in which a cell contains the information of the set of entities that may alternatively occupy the cell and the probability of this occupancy. Thus, each cell <*x*,*y*,*z*> of the metric map of an agent *i* is set to a set of pairs $\phi^i_{x,y,z} = \{< p^i_1, E^i_1 >, < p^i_2, E^i_2 >, ..., < p^i_{n_i}, E^i_{n_i} >, < p^i_{n_{i+1}}, 0 >\}$, where E^i_j is the identifier of the j^{th} entity that may occupy the cell <*x*,*y*,*z*> of the metric map of agent *i* with probability $p^i_j \in [0,1]$, and such that $\sum_{j=1}^{n_{i+1}} p^i_j = 1$. Note that the pair $< p^i_{n_{i+1}}, 0 >$ is included in order to express the probability of the cell being empty. Cells that are completely unknown, i.e., for which there are not yet any assumptions/expectations about their occupancy, are set with an empty set of pairs $\phi^i_{x,y} = \{\}$. Note also that each entity may occupy more than a single cell, i.e., there might be several adjacent cells with the same E^i_j.

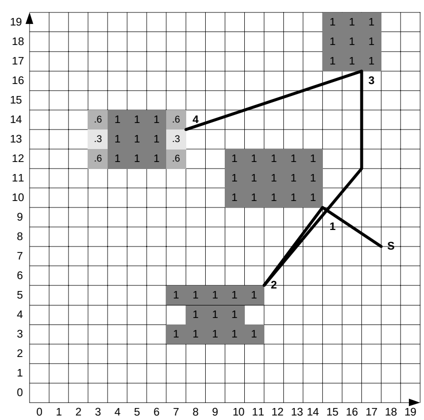

Fig. 1. An example of a metric map. Although metric maps are of three-dimensional kind, for the sake of simplicity, it is represented here only in two dimensions. For the same reason the identifier of the entities are not represented. The path followed by the agent to explore this environment (comprising buildings) is also depicted

2.2 Memory for Entities

The set of descriptions of entities perceived from the environment are stored in the *episodic memory of entities*. Each one of these descriptions is a case of the form *<ID,PS,F>*, where *ID* is a number that uniquely identifies the entity in the environment, *PS* is the physical structure, and *F* is the function of the entity [15]. The sensors may provide incomplete information about an entity (for instance, only part of the physical structure may be seen or the function of the entity may be undetermined). In this case the missing information is filled in by making use of Bayes' rule [34], i.e., the missing information is estimated taking into account the available information and cases of other entities previously perceived and already stored in the *episodic memory of entities*. This means some of the descriptions of entities stored in memory are uncertain or not completely known (e.g., element 4 of Fig. 2).

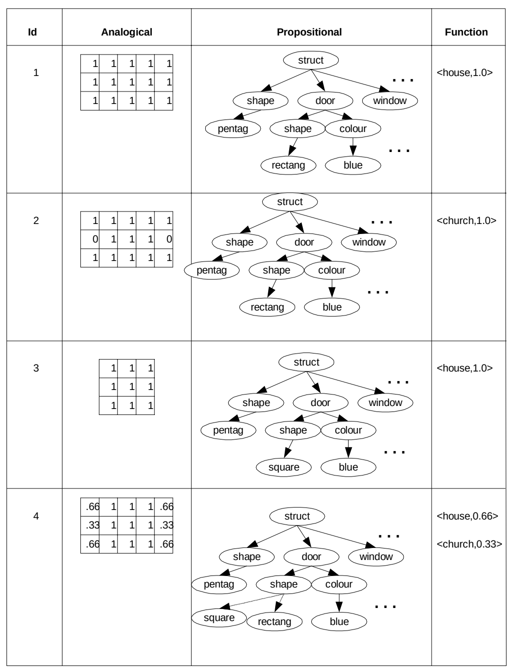

Fig. 2. Example of the episodic memory of entities in the domain of buildings. Although the matrix of the analogical description is of three-dimensional kind, for the sake of simplicity, it is represented here as a two-dimensional matrix corresponding to the upper view of the entity

The physical structure of an entity may be described analogically or propositionally [1, 14]. The analogical representation reflects directly the real physical structure while the propositional representation is a higher level description (using propositions) of that real structure.

The analogical description of the physical structure of an entity comprises a three-dimensional matrix and the coordinates of the centre-of-mass relative to the entity and to the environment spaces. Notice that the three-dimensional matrix of the entity is a submatrix of the matrix that represents the metric map.

The propositional description of the physical structure of an entity relies on the representation through semantic features or attributes much like in semantic networks or schemas [1]. According to this representation approach, entities are described by a set of attribute-value pairs that can be represented in graph-based way [24].

The function is simply a description of the role or category of the entity in the environment. For instance, a house, a car, a tree, etc. Like the description of the physical structure, this may be probabilistic because of the incompleteness of perception. This means, this is a set $F = \{<function_i, prob_i>$: i=1,2, …, n, where n is the number of possible functions and P(“function” = $function_i$) = $prob_i\}$.

Concrete entities (i.e., entities represented in the episodic memory) with similar features may be generalized or abstracted into a single one, an abstract entity, which is stored in the *semantic memory for entities*. Fig. 3 presents a semantic memory obtained from the episodic memory of entities shown in Fig. 2.

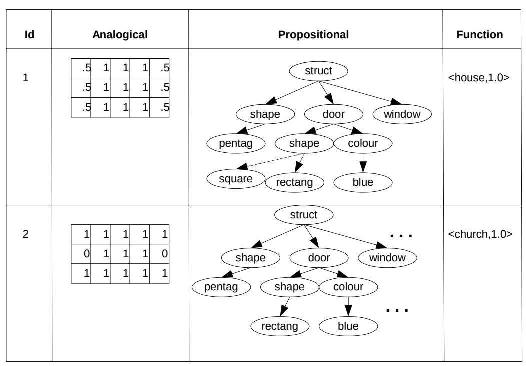

Fig. 3. Example of the semantic memory of entities

2.3 Memory for Plans

Like entities, we may distinguish two main kinds of plans: concrete plans, i.e., cases of plans, and abstract plans (e.g., [5]). Concrete plans and abstract plans are interrelated since concrete plans are instances of abstract plans and these are built from concrete plans.

We represent plans as a hierarchy of tasks (a variant of HTNs (e.g., [13]) (see Fig. 4). Formally, a plan is a tuple $AP = <T, L>$, where T is the set of tasks and L is the set of links. This structure has the form of a planning tree [23], i.e., it is a kind of AND/OR tree that expresses all the possible ways to decompose an initial task network. Like in regular HTNs, this hierarchical structure of a plan comprises primitive tasks or actions (non-decomposable tasks) and non-primitive tasks (decomposable or compound tasks). Primitive tasks correspond to the leaves of the tree and are directly executed by the agent, while compound tasks denote desired changes that involve several subtasks to accomplish it. For instance, the leaf node *PTRANS* of Fig. 4 is a primitive task, while *visitEntity* is a compound task. A task *t* is both conditional and probabilistic (e.g., [8]). This means each task has a set of conditions C={ c_1, c_2, ..., c_m} and for each one of these mutually exclusive and exhaustive conditions, c_i, there is a set of alternative effects ε^i={$< p_1^i, E_1^i >$, $< p_2^i, E_2^i >$, ..., $< p_{n_i}^i, E_{n_i}^i >$}, where E_j^i is the j^{th} effect triggered with probability $p_j^i \in [0,1]$ by condition c_i (i.e., $P(E_j^i \mid c_i) = p_j^i$), and such that $\sum_{j=1}^{n_i} p_j^i = 1$. Each effect contains information about changes produced in the world by achieving the goal task. Thus, an effect may give information about the amount of power consumed, the new location of the agent, the emotions felt, etc.

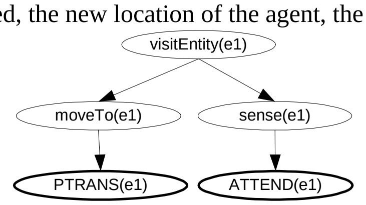

Fig. 4. Example of simple plan. Primitive tasks are represented by thick ellipses while non-primitive tasks are represented by thin ellipses

3 Exploration using CBR

Each agent is continuously performing the following deliberative reasoning/decision-making algorithm. Each agent at a given time senses the environment to look for entities and compute the current world state (location, structure and function of those entities) based on the sensorial information and on the generation of expectations for the missing information. The result is a set of cases of entities, each one describing an entity that was perceived. Then, the episodic memory and metric map are updated based on these episodic entities. New intentions/goals of kind *visitEntity* are generated for each unvisited entity within the visual range based on the goal tasks of cases of past plans. In addition, a goal of the kind *visitLoc* is generated for some frontier cells [41] (another possible kind of goal is *rechargeBattery*). These goals are then ranked according to their Expected Utility (EU) [33], which is computed based on the estimated intensities for the motivations that they may elicit as explained below [27]. The first one in the ranking, i.e., the goal with the highest EU is taken and a HTN plan is generated based on cases of past plans. Then, the agent executes this plan.

We will now describe in more detail the steps related with the generation of assumptions/expectations and generation of agent's goals and respective plans.

3.1 Case-Based Generation of Assumptions/Expectations

As we said before, it is very difficult for an agent to get all the information about the surrounding environment. One reason is that the perceptual information is incomplete. However, taking as evidence the available information it is possible to generate expectations/assumptions for the missing information using a Bayesian approach [34]. Actually, Bayes' rule, represented as follows, may be used:

$$P(H_i \mid E_1, E_2, ..., E_m) = \frac{P(E_1 \mid H_i) \times P(E_2 \mid H_i) \times ... \times P(E_m \mid H_i) \times P(H_i)}{\sum_{l=1}^{n} P(E_1 \mid H_l) \times P(E_2 \mid H_l) \times ... \times P(E_m \mid H_l) \times P(H_l)} \quad \textbf{(1)}$$

where E_1, E_2, …, E_m are pieces of evidence, i.e., the available information, and H_i, i=1,2,…,n, are mutually exclusive and collectively exhaustive hypotheses (retrieved from past cases of entities) for a specific piece of the missing information. Each conditional probability $P(E|H)$ is given by the number of times E and H appeared together in the cases of entities stored in memory divided by the number of times H appeared in those case of entities (when E and H have never appeared together $P(E|H)= P(E)$). In our work the evidence is the description (propositional) of the physical structure of the entities such as their shape (rectangular, squared, etc.), shape of their constituent parts (in case there are any), color, etc. The hypotheses could be not only for parts of the descriptions of the physical structure but also for the function or category of the entity. In this case, the result is a probability distribution for the function of the entity (e.g., *P(Function=house)*=0.666; *P(Function=church)*=0.333). Based on this distribution, the analogical description of the entity may be now estimated taking into account the analogical descriptions of the entities with these functions. This means that we are considering the reference class as comprising the entities with the same function. Notice that this resulting analogical description is probabilistic. Thus, for instance, considering the semantic memory presented in Fig. 3 and the probability distribution for the function of an entity [*P(Function=house)*=0.66, *P(Function=church)*=0.33], the resulting analogical description is similar to that of entity 4 of the episodic memory depicted in Fig. 2. This is computed as follows. For all function X: (i) take the analogical description of each possible entity with function *X* and multiply the occupancy value of each cell by *P(Function=X)*; (ii) superimpose the analogical descriptions obtained in the previous step summing the occupancy values of the superimposed cells.

3.2 Case-Based Generation of Goals and Plans

The algorithm for the generation and ranking of goals/intentions (Fig. 5) is as follows. First, the set of different goal tasks present in the memory of plans are retrieved and, for each kind, a set of new goals is generated using the following procedure: given a goal task retrieved from a plan in the memory of plans, the memory and the perception of the agent, similar goals are generated by adapting the past goal to situations of the present state of the world. The adaptation strategies used are mainly substitutions [20]. Thus, for instance, suppose the goal task *visitEntity*(*e7*) is present in the memory of the agent. Suppose also that the agent has just perceived three entities present in the environment, *e1*, *e2* and *e3*. The entity to which *visitEntity* is applied (*e7*) may be substituted by *e1*, *e2* or *e3*, resulting in three new goals: *visitEntity*(*e1*), *visitEntity*(*e2*), *visitEntity*(*e3*). Then, the EU of each goal task is computed. As said above, a task *T* is both conditional and probabilistic (e.g.: [8]). Thus, the execution of a goal task under a given condition may be seen according to Utility Theory as a lottery [33]:

$$Lottery(T) = \left[p^1 \times p_1^1, E_1^1; p^1 \times p_2^1, E_2^1; ...; p^m \times p_{n_m}^m, E_{n_m}^m \right] \quad \textbf{(2)}$$

, where p^i is the probability of the condition c_i, p_j^i is the probability of the j^{th} effect, E_j^i, of condition c_i.

The EU of *T* may be then computed as follows:

$$EU(T) = \sum_{k,j} p^k \times p_j^k \times EU(E_j^k) \quad \textbf{(3)}$$

The computation of $EU(E_j^k)$ is performed predicting the motivations that could be elicited by achieving/executing the goal task [11, 32]. We confined the set of motivations to those that are more related with exploratory behaviour in humans [6]. Thus, the intensities of surprise, curiosity and hunger felt by the agent when the effect takes place are estimated based on the information available in the effect about the changes produced in the world or based on the intensities of emotions and other motivations felt in past occurrences of the effect of the task.

Surprise is given by [26]:

$$SURPRISE(Agt, Obj_k) = UNEXPECTEDNESS(Obj_k, Agt(Mem)) = 1 - P(Obj_k) \quad \textbf{(4)}$$

, where Obj_k is the direct object of task *T* when E_j^k takes place, i.e., the entity that is visited.

Curiosity is computed as follows:

$$CURIOSITY(Agt, Obj_k) = DIFFERENCE(Obj_k, Agt(Mem)) \quad \textbf{(5)}$$

The measure of difference relies heavily on error correcting code theory [17]: the function computes the distance between two entities represented by graphs, counting the minimal number of changes (insertions and deletions of nodes and edges) required to transform one graph into another (for a similar approach see [31]).

The hunger drive is defined as the need of a source of energy. Given the capacity *C* of the storage of that source, and *L* the amount of energy left ($L \leq C$), the hunger elicited in an agent is computed as follows:

$$HUNGER(Agt) = C - L \quad \textbf{(6)}$$

The following function is used to compute $EU(E_j^k)$:

$$EU(E_j^k) = \frac{\alpha_1 \times U_{surprise}(E_j^k) + \alpha_2 \times U_{curiosity}(E_j^k) + \alpha_3 \times U_{hunger}(E_j^k)}{\sum_i \alpha_i} = \tag{7}$$

$$= \frac{\alpha_1 \times Surprise(E_j^k) + \alpha_2 \times Curiosity(E_j^k) + \alpha_3 \times Hunger(E_j^k)}{\sum_i \alpha_i}$$

, where, α_3 = -1 and α_i ($i \neq 3$) may be defined as follows:

$$\alpha_i = \begin{cases} 1 \Leftarrow C - HUNGER(Agt) - D > 0 \\ 0 \Leftarrow otherwise \end{cases} \tag{8}$$

, where *D* is the amount of energy necessary to go from the end location of goal task *T* to the closer place where energy could be recharged, and *C* is the maximum amount of energy that could be stored by the agent. The functions $Surprise(E_j^k)$, $Curiosity(E_j^k)$ and $Hunger(E_j^k)$ are replaced by the functions of curiosity, surprise and hunger defined above and applied for the entities perceived when the effect E_j^k takes place.

The surprise and curiosity of an effect of a task are elicited by the entities that the agent perceives.

Algorithm generateRankGoals(*newRankedGoals*)
Output: *newRankedGoals* – the set of ranked goals
newGoals ← ∅
setPastGoals ← {*x*: *x* is a goal task belonging to some plan in memory}
for each *goal* **in** *setPastGoals* **do**
    adaptationGoal←adaptGoal(*goal*,*agtMemory*,*agtPercepts*)
    newGoals ← *newGoals* ∪ *adaptationGoals*
end for each
for each *goal* **in** *newGoals* **do**
$EU(T) = \sum_{k,j} p^k \times p_j^k \times EU(E_j^k)$
end for each
insert(*goal*,*newRankedGoals*)
return *newRankedGoals*
end

Fig. 5. Algorithm for the case-based generation of goals

This dependence of the parameters α_i ($i \neq 3$) on the hunger of the agent partially models the results of Berlyne's experiments (e.g., [6]) that have shown that in the absence of (or despite) known drives, humans tend to explore and investigate their environment as well as seek stimulation. Actually, surprise and curiosity are taken into account to compute the EU of a task only when there is enough energy to go from the end location of goal task *T* to the closest place where an energy source could be found. Otherwise, only hunger is taken into account for the EU of tasks and further ranking. This means that in this situation (when hunger is above a specific threshold), only the goal of *rechargeBattery* has an EU > 0. In the other situations (hunger below a specific threshold), hunger plays the role of a negative reward decreasing the utility of a task by the percentage of energy needed after the task is completed. Thus, the more the distance to the location after the execution of a task the more the energy required and the less the utility of that task.

However, the environment is not confined to entities. It might have regions that are not yet explored. Therefore, goals of kind *visitLoc* are also retrieved from past plans and adapted for the current frontier cells. Not all the cells are considered. We followed an approach similar to [9], i.e., different target cells are assigned to each agent so that the overlapped area of the visual fields of the agents in those cells is minimized. The EU of a goal task of this kind is also computed with the above equation 3, although in this case curiosity is computed based on the estimation of the amount of unknown cells inside the visual field if the agent is at the destination location. Surprise is assumed to be 0.

A HTN plan is generated for the first goal in the ranking as follows. A problem is an initial and incomplete HTN, i.e., a set of goal tasks. Planning is a process by which that initial HTN is completed resulting in an abstract plan ready to be executed and incorporating alternative courses of action, i.e., it includes replanning procedures. Roughly speaking, this involves the following steps: first, the structure of the abstract plan (HTN) is built based on cases of past plans (this is closely related to the regular HTN planning procedure); then the conditional effects, probabilities as well as the EU are computed for the primitive tasks of this abstract plan based on the primitive tasks of cases of past plans; finally, these properties (conditional effects and respective probabilities, and EU) are propagated upward in the HTN, from the primitive tasks to the main task of the HTN. Fig. 6 presents this algorithm and Fig. 7 illustrates the process of building the structure of a plan using an AND/OR decomposition approach. For more details about the algorithm for constructing an abstract plan see [29].

Algorithm CONSTRUCT-ABSTRACT-PLAN(*abstPlan*)
  abstPlan ← BUILD-STRUCTURE(*abstPlan*)
  primTasks ← GET-PRIM-TASKS(*abstPlan*)
  primTasksAllPlanCases← GET-PRIMTASKS-ALL-PLAN-CASES()
  COMPUT-PRIMTASKS-PROPS(*primTasks*,*primTasksAllPlanCases*)
  abstPlan←PROPAGAT-PROPS-UPWARD(*primTasks*,*abstPlan*)
  return *abstPlan*
end

Fig. 6. Algorithm for constructing an abstract plan

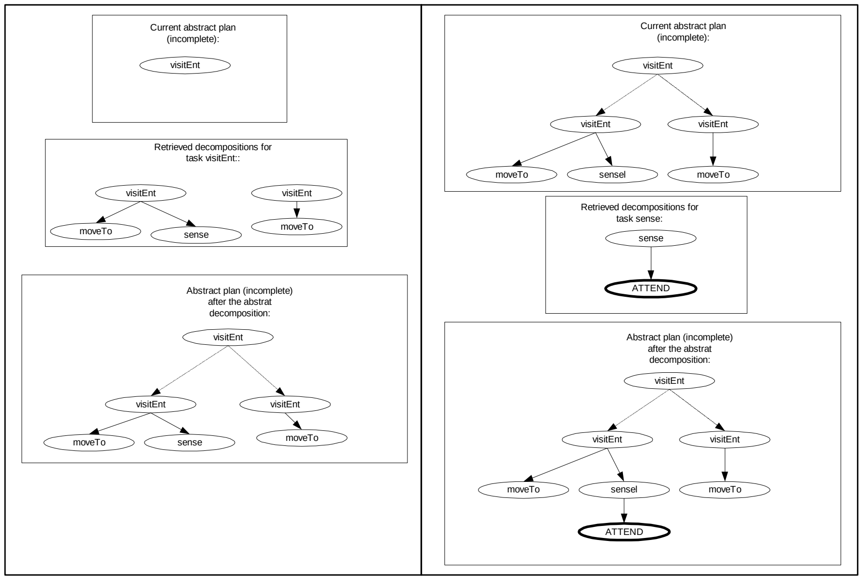

Fig. 7. Two illustrative examples of the process of building the structure of a plan using an AND/OR decomposition approach

4 Experiment

We conducted an experiment in a simulated environment comprising buildings in order to evaluate the role of the size of the case-base of entities on the exploration performance of an agent. To do so, we ran an agent in the same environment (see Fig. 8) with different starting case-bases of entities. These memories ranged from 1 to 5 cases. All of these cases were built from 5 entities, selected randomly among the 10 entities that populate the environment. For instance, the case-base of size 3 comprised the cases 1, 2 and 3 described in Fig. 2, while the case-base of size 1 comprised only case 1, and the case-base of size 2 was formed with cases 1 and 2. The other larger case-bases of size 4 and 5 comprised in addition cases of entities 4 and 5, respectively. We then let the agent explore the environment during a limited time so that it can't explore exhaustively the whole environment. Actually, this time limit was defined so that it can't even visit any entity (stopped at position *S*). Therefore, the agent had to build the map of the environment by generating assumptions/expectations for the unvisited entities as described in section 3.1. Finally, we compared these maps built by the agent with the real map (the map that should had been built by an ideal agent). The difference or inconsistency between two maps was measured summing the difference between the occupancy values of any two correspondent cells (cells with similar coordinates) of the two maps. Fig. 9 presents the results of the experiment. As it can be seen the map inconsistency decreases monotonically with the increasing size of the case-base. This can be explained as follows. The entities are only perceived at a certain distance. For these entities, the cases generated are probabilistic (especially their analogical physical description). With larger case-bases the agent is able to generate more accurate probabilistic analogical descriptions for the entities, i.e., the expectations are closer to the reality. However, this higher accuracy is achieved by a slower reasoning process because the agent has to take into account more cases. This is related with the Utility Problem [37] since it is expected that the increase of the size of the case-base is of significant benefit only up to a certain size. Case-bases larger than that point are expected to include redundant cases. Case base maintenance techniques (e.g., [21]) might be applied in order to avoid this.

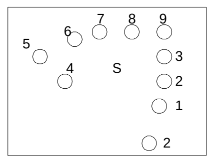

Fig. 8. Environment with 10 entities (two of them, entities with identifier 2, are identical). Entities 1, 2 and 3 are those described in the episodic memory of Fig. 2

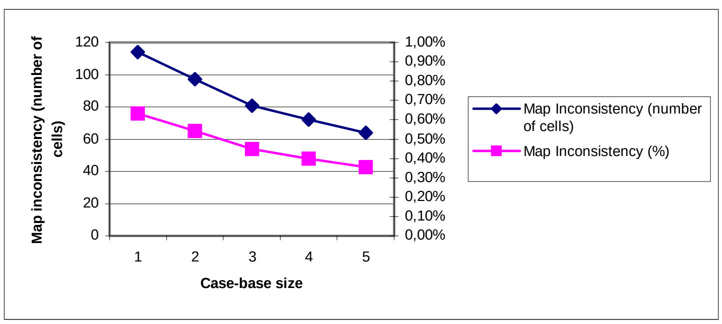

Fig. 9. Inconsistency between real and built maps represented from two points of view: number of inconsistent cells; percentage of inconsistency (computed dividing the number of inconsistent cells by the total amount of cells in the three-dimensional environment – in this case this was 18000 cells)

5 Discussion and Conclusions

We presented a case-based approach for the exploration of unknown environments. The experiment conducted allow us to conclude that the exploration performance may be improved by previously training the agent in similar environments so that case-bases of entities and plans are acquired. The main advantage is that agents are able to build more accurate maps of the world when using larger case-bases especially when they can't explore exhaustively the whole environment. Actually, the agent does not have to explore all the regions of the environment, such as the invisible side of the entities, since it is able to predict that inaccessible information. However, too much larger case-bases do not improve significantly the exploration performance relative to medium size case-bases because they imply higher computation times and therefore the exploration task is delayed because fewer entities are visited or perceived. The disadvantage of this approach is that the built maps may be more inconsistent than those acquired from an exhaustive exploration of the environment. However, this inconsistency is almost insignificant since, for instance, for a case-base with a single case, in the three-dimensional environment of the experiment with 18000 cells, the inconsistency was 114 cells, i.e., 0.63% of the environment. However, this depends on the kind of cases stored in the case-base as well as on the complexity of the environment (for a similar experience with different environments see [30]; for another experiment about the trade-off between exploration and exploitation see [28]).

Acknowledgments

The PhD of Luís Macedo is financially supported by PRODEP III.

References

[1] A. Aitkenhead and J. Slack, *Issues in cognitive modelling*. London: Lawrence Erlbaum Associates, 1987.

[2] J. Amat, R. Màntaras, and C. Sierra, "Cooperative autonomous low-cost robots for exploring unknown environments," in *Proccedings of the International Symposium on Experimental Robotics*, 1995.

[3] D. Anguelov, R. Biswas, D. Koller, B. Limketkai, S. Sanner, and S. Thrun, "Learning hierarchical object maps of non-stationary environments with mobile robots," in *Proceedings of the 17th Annual Conference on Uncertainty in AI*, 2002.

[4] D. Apostolopoulos, L. Pedersen, B. Shamah, K. Shillcutt, M. Wagner, and W. Whittaker, "Robotic antarctic meteorite search: outcomes," in *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, 2001, pp. 4171-4179.

[5] R. Bergmann and W. Wilke, "On the Role of Abstraction in Case-Based Reasoning," in *Advances in Case-Based Reasoning - Proceedings of the Third European Workshop on Case-Based Reasoning*, vol. 1168, *Lecture Notes in Artificial Intelligence*, I. Smith and B. Faltings, Eds. Berlin: Springer Verlag, 1996, pp. 28-43.

[6] D. Berlyne, "Novelty and curiosity as determinants of exploratory behavior," *British Journal of Psychology*, vol. 41, pp. 68-80, 1950.

[7] R. Biswas, B. Limketkai, S. Sanner, and S. Thrun, "Towards object mapping in non-stationary environments with mobile robots," in *Proceedings of the Conference on Intelligent Robots and Systems*, 2002.

[8] J. Blythe, "Decision-Theoretic Planning," *AI Magazine, Summer 1999*, 1999.

[9] W. Burgard, D. Fox, M. Moors, R. Simmons, and S. Thrun, "Collaborative Multi-Robot Exploration," in *Proceedings of the IEEE International Conference on Robotics and Automation*: IEEE, 2000.

[10] W. Burgard, M. Moors, and F. Schneider, "Collaborative exploration of unknown environments with teams of mobile robots," in *Advances in plan-based control of robotic agents*, vol. 4266, *Lectures Notes in Computer Science*, M. Beetz, J. Hertzberg, M. Ghallab, and M. Pollack, Eds. Berlin: Springer Verlag, 2002.

[11] C. Castelfranchi, R. Conte, M. Miceli, and I. Poggi, "Emotions and goals," in *Perspectives on Cognitive Science*, B. Kokinov, Ed. Sofia: New Bulgarian University, 1996, pp. 131-145.

[12] H. Endres, W. Feiten, and G. Lawitzky, "Field test of a navigation system: autonomous cleaning in supermarkets," in *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, 1998.

[13] K. Erol, J. Hendler, and D. Nau, "UMCP: A sound and complete procedure for hierarchical task-network planning," in *Proceedings of the International Conference on AI Planning Systems*, 1994, pp. 249-254.

[14] M. Eysenck and M. Keane, *Cognitive psychology*. London: Lawrence Erlbaum Associates, 1991.

[15] A. Goel, "Representation of Design Functions in Experience-Based Design," in *Intelligent Computer Aided Design*, D. Brown, M. Walderon, and H. Yosnikawa, Eds.: Elsevier Science, 1992.

[16] D. Goldberg, V. Cicirello, M. Dias, R. Simmons, S. Smith, T. Smith, and A. Stentz, "A Distributed Layered Architecture for Mobile Robot Coordination: Application to Space Exploration," in *3rd International NASA Workshop on Planning and Scheduling for Space*. Houston, 2002.

[17] R. Hamming, "Error Detecting and Error Correcting Codes," *The Bell System Technical Journal*, vol. 26, pp. 147-60, 1950.

[18] Y. Huang, Z. Cao, S. Oh, E. Kattan, and E. Hall, "Automatic operation for a robot lawn mower," in *SPIE Conference on Mobile Robots*, vol. 727, 1986, pp. 344-354.

[19] C. Izard, *The Psychology of Emotions*. NY: Plenum Press, 1991.

[20] J. Kolodner, *Case-Based Reasoning*. San Mateo, CA: Morgan-Kaufmann, 1993.

[21] D. Leake, Wilson, D., "Categorizing case-base maintenance: dimensions and directions," in *Proceedings of the 4th European Workshop on Case-Based Reasoning*. Berlin: Springer-Verlag, 1998.

[22] D. Lee and M. Recce, "Quantitative evaluation of the exploration strategies of a mobile robot," *AAAI-94*, 1994.

[23] A. Lotem and D. Nau, "New advances in GraphHTN: Identifying independent subproblems in large HTN domains," in *Proceedings of the International Conference on AI Planning Systems*, 2000, pp. 206-215.

[24] L. Macedo and A. Cardoso, "Nested-Graph structured representations for cases," in *Advances in Case-Based Reasoning - Proceedings of the 4th European Workshop on Case-Based Reasoning*, vol. 1488, *Lecture Notes in Artificial Intelligence*, B. Smyth and P. Cunningham, Eds. Berlin: Springer-Verlag, 1998, pp. 1-12.

[25] L. Macedo and A. Cardoso, "SC-EUNE - Surprise/Curiosity-based Exploration of Uncertain and Unknown Environments," in *Proceedings of the AISB'01 Symposium on Emotion, Cognition and Affective Computing*. York, UK: University of York, 2001, pp. 73-81.

[26] L. Macedo and A. Cardoso, "Modelling Forms of Surprise in an Artificial Agent," in *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, J. Moore and K. Stenning, Eds. Mahwah, NJ: Erlbaum, 2001, pp. 588-593.

[27] L. Macedo and A. Cardoso, "A Motivation-based Approach for Autonomous Generation and Ranking of Goals in Artificial Agents," in *Proceedings of the AISB'04 Fourth Symposium on Adaptive Agents and Multi-Agent Systems*, D. Kudenko, Ed. Leeds: SSAISB, 2004.

[28] L. Macedo and A. Cardoso, "Exploration of Unknown Environments with Motivational Agents," in *Proceedings of the Inhird International Joint Conference on Autonomous Agents and Multiagent Systems*, N. Jennings and M. Tambe, Eds.: ACM Press, 2004.

[29] L. Macedo and A. Cardoso, "Case-Based, Decision-Theoretic, HTN Planning," in *Proceedings of the 7th European Conference on Case-Based Reasoning*, P. Calero and P. Funk, Eds. Berlin: Springer, 2004.

[30] L. Macedo and A. Cardoso, "Building maps from incomplete environment information: a cognitive approach based on the generation of expectations," in *Proceedings of 5th IFAC Symposium on Intelligent Autonomous Vehicles*, M. Ribeiro, Ed. Oxford, UK: Elsevier Science, 2004.

[31] B. Messmer and H. Bunke, "A new algorithm for error-tolerant subgraph isomorphism detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, 1998.

[32] R. Reisenzein, "Emotional Action Generation," in *Processes of the molar regulation of behavior*, W. Battmann and S. Dutke, Eds. Lengerich: Pabst Science Publishers, 1996.

[33] S. Russel and P. Norvig, *Artificial Intelligence - A Modern Approach*. Englewood Cliffs, NJ: Prentice Hall, 1995.

[34] G. Shafer and J. Pearl, "Readings in Uncertain Reasoning". Palo Alto, CA: Morgan Kaufmann, 1990.

[35] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes, "Coordination for Multi-Robot Exploration and Mapping," in *Proceedings of the AAAI-2000*, 2000.

[36] M. Simoncelli, G. Zunino, H. Christensen, and K. Lange, "Autonomous pool cleaning: self localization and autonomous navigation for cleaning," *Journal of Autonomous Robots*, vol. 9, pp. 261-270, 2000.

[37] B. Smyth and P. Cunningham, "The utility problem analysed," in *Advances in Case-Based Reasoning - Proceedings of the Third European Workshop on Case-Based Reasoning*, vol. 1168, *Lecture Notes in Artificial Intelligence*, I. Smith and B. Faltings, Eds. Berlin: Springer Verlag, 1996, pp. 392-399.

[38] S. Thrun, "Efficient exploration in reinforcement learning," Carnegie Mellon University, Computer Science Department, Pittsburgh, PA CMU-CS-92-102, 1992.

[39] S. Thrun, "Exploration in active learning," in *Handbook of Brain Science and Neural Networks*, M. Arbib, Ed., 1995.

[40] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. San Mateo, CA: Morgan Kaufmann, 2002.

[41] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proceedings of the Second Iinnternational Conference on Autonomous Agents*. Minneapolis, MN, 1998.