Integrating a High-Level Planner with a Low-Level Robot

Kartik Talamadupula
Department of Computer Science
Arizona State University
Tempe, AZ 85287 USA
krt@asu.edu
Advised by: Subbarao Kambhampati
Joint work with: J. Benton, Paul Schermerhorn, Matthias Scheutz

Abstract

The theoretical aspects of automated planning, and associated results, have for long outstripped the application of these methods in problems of relevance in the real-world. Even today, advances in planning-related theory are reported on a far more regular basis than results on integrating and using these methods in applications. One major reason for this discrepancy is that the state-of-the-art planning systems that are used to report theoretical advances are hard to integrate and use in application scenarios. In this paper, we present work on an integrated planning and robotic architecture that actively directs an agent engaged in an urban search and rescue scenario. We describe three salient features that comprise the planning component of this system, namely (1) planning in a world open with respect to objects, (2) execution monitoring and replanning abilities, and (3) handling soft goals, and detail the interaction of this trio in representing and solving the application scenario at hand.

Introduction

In recent years, considerable progress has been made in developing highly efficient planners. While the class of problems handled by these planners is expanding every year, most of them are still evaluated in *closed-world* conditions, where the planner has knowledge of all the objects in the world. In many real world applications, such closed-world assumptions do not hold. A case in point is robots exploring a partially known world, such as in a search and rescue scenario. An important challenge for the planning community is to understand what aspects of current planning systems are relevant for handling such scenarios and what extensions need to be made. This is precisely the broader aim of the current paper. Before elaborating on the specifics of our investigation, we begin by giving the details of our application scenario.

We consider a human-robot team engaged in an urban search and rescue (USAR) scenario inside a building of interest. The robot is placed at the beginning of a long corridor, while the human team member (who has intimate knowledge of the building's layout) removed from the scene and can only interact with the robot via on-board wireless audio communication. The corridor in which the robot is located has doors leading off from either side into rooms, a fact known to the robot. However, unknown to the robot (and the human team member) initially is the possibility that these rooms may contain injured humans (victims). The robot is initially given a hard goal of reaching the end of the corridor by a given deadline based on wall-clock time. As the robot executes a plan to achieve that goal, the team is given additional information regarding victims being in rooms. Also specified with this information is a new soft goal, to report the location of victims.

The fallibility of human experts in completely specifying information relevant to the given problem and goals up-front makes it quite likely that knowledge needed to achieve some soft goals may be specified at some later stage during the planning process. This partial knowledge may be unbounded, existing in both the problem dynamics and objectives. In the USAR scenario, for example, the knowledge that injured people are in rooms may be relayed to the planner while it is engaged in planning for the executing robot. In order to handle the specification of such statements in the midst of an active planning process, and enable the use of knowledge thus specified, we need to relax two crucial assumptions that most modern planners rely on. The first is the closed world assumption with respect to the constants (objects) in the problem – the planner can no longer assume that the only objects and facts in the scenario are those specified in the initial state alone. We must also interleave planning with execution monitoring and, if required, replanning in order to account for the new information.

Planning in an Open World

We use *SapaReplan* (Cushing, Benton, and Kambhampati 2008), a state-of-the-art planner that (like most current planners) operates in a closed world by assuming that all objects and facts related to those objects are known up-front in the initial state. In the rest of this paper, we use *closed world assumption* to indicate this. There exists an obvious problem with making such an assumption within an open world environment. The planner does not have complete a priori knowledge of all objects (e.g., injured people) – because of this, we must consider general, quantified goals while at the same time allowing for the discovery of new objects that enable the achievement of goals. This combination shows the inherent connection between sensing and goal achievement – some goals only exist given particular facts whose truth value remains unknown at the initial state. It further highlights a strict need for sensing in order to ensure high reward for a given plan (in terms of goal achievement). On top of this, we have a set of objects that imply certain facts; for example, a door implies the existence of a room and hence the potential for goal (and reward) achievement.

Interleaving Planning and Execution

For most of the sensors on the robot, it is too expensive to sense at every step, so knowing exactly when to engage in perceptual monitoring is of critical importance. Planning through an open world also introduces the possibility of dangerous faults or nonsensical actions. While in some sense, this risk can be quantified with a measure (see Garland and Lesh (2002), for example), indicating the risk of a plan does nothing to address those risks. A more robust approach in an online scenario involves *planning to sense* in a goal-directed manner.

Problem Updates New sensory information can be sent to the planner at any time, either during planning or after a plan has been output. New data can become known from other sources as well (e.g., a commander may issue a new goal or give new facts about the world). Regardless of the originating source, the monitor receives updates from the *goal manager* and correspondingly modifies the planner's representation of the problem. Updates can include new objects, timed events (i.e., an addition or deletion of a fact at a particular time, or a change in a numeric value such as action cost), the addition or modification (on the deadline or reward) of a goal, and a time point to plan from. Updates from the world consist of some number of: (1) new objects, (2) exogenous events, (3) new or updated goals, and (4) the current time.

Partial Satisfaction Planning A Partial Satisfaction Planning (PSP) problem involves actions and (soft) goals with varying costs and rewards. This contrasts with classical planning, which focuses on hard goal achievement. The planning objective is to find plans with high *net benefit* (cumulative goal reward minus plan action cost) by considering which goals should be achieved and which should be ignored due to their high cost or other resource constraints (such as time). The selection process occurs during an A* search. At each search state, the planner's heuristic evaluates the cost for achieving individual goal facts and removes those goals (and supporting actions) that appear too costly to achieve. That is, a goal will not be pursued at a given state if the estimated cost of achievement outweighs the reward.

Goals in an Open World

To handle the issues inherent with specifying information critical to goal achievement in an open world, we introduce a novel construct called an *open world quantified goal* (OWQG) (Talamadupula et al. 2010) that combines information about objects that *may be* discovered during execution with partial satisfaction aspects of the problem. Using an OWQG, the domain expert can furnish details about what new objects may be encountered through sensing and include goals that relate directly to the sensed objects. This can be seen as a complementary approach to handling open world environments using *local closed world* (LCW) information produced by sensing actions (Etzioni, Golden, and Weld 1997).

An OWQG is a tuple $\mathcal{Q} = \langle F, \mathcal{S}, \mathcal{P}, \mathcal{C}, \mathcal{G} \rangle$ where F and $\mathcal{S}$ are typed variables that are part of the problem Π, where F belongs to the object type that $\mathcal{Q}$ is quantified over, and $\mathcal{S}$ belongs to the object type about which information is to be sensed. $\mathcal{P}$ is a predicate which ensures sensing closure for every pair $\langle f, s \rangle$ such that f is of type F and s is of type $\mathcal{S}$, and both f and s belong to the set of objects in the problem, $\mathcal{O} \in \Pi$; for this reason, we term $\mathcal{P}$ a *closure condition*. $\mathcal{C} = \bigwedge_i c_i$ is a conjunctive first-order formula where each c_i is a statement about the openness of the world with respect to the variable S. For example, `c = (in ?hu - human ?z - zone)` with `S = ?hu - human` means that c will hold for new objects of the type 'human' that are sensed. Finally, $\mathcal{G}$ is a quantified goal on $\mathcal{S}$.

Of the components that make up an open world quantified goal $\mathcal{Q}$, $\mathcal{P}$ is required[1] and F and S must be non-empty, while the others may be empty. If $\mathcal{G}$ is empty, i.e., there is no new goal to work on, the OWQG $\mathcal{Q}$ can be seen simply as additional knowledge that might help in reasoning about other goals. Newly discovered objects may enable the achievement of goals, granting the opportunity to pursue reward. For example, detecting a victim in a room will allow the robot to report the location of the victim (where reporting gives reward). Given that the reward in our scenario is for each reported injured person, there exists a quantified goal that must be allowed partial satisfaction. In other words, the universal base, or total grounding of the quantified goal on the real world, may remain unsatisfied while its component terms may be satisfied. To handle this, we use partial satisfaction planning (PSP) (van den Briel et al. 2004), where the objective is to maximize the difference between the reward given to goals, and the cost of actions. Reward is associated with each term $g \in \mathcal{G}$ satisfied, $u(\mathcal{G})$. Additionally each term g is considered soft in that it may be "skipped over" and remain unachieved.

Implementation

To handle open world quantified goals, the planner grounds the problem into the closed world using a process similar to Skolemization. More specifically, we generate *runtime objects* from the sensed variable $\mathcal{S}$ that explicitly represent the potential existence of an object to be sensed. These objects are marked as system generated runtime objects. Given an OWQG $\mathcal{Q} = \langle F, \mathcal{S}, \mathcal{P}, \mathcal{C}, \mathcal{G} \rangle$, one can look at $\mathcal{S}$ as a Skolem function of F, and runtime objects as Skolem entities that substitute for the function. Runtime objects are then added to the problem and ground into the closure condition $\mathcal{P}$, the conjunctive formula $\mathcal{C}$, and the open world quantified goal $\mathcal{G}$. Runtime objects substitute for the existence of S dependent upon the variable F. The facts generated by following this process over $\mathcal{C}$ are included in the set of facts in the problem

[1] If $\mathcal{P}$ were allowed to be empty, the planner could not gain closure over the information it is sensing for, which will result in it directing the robot to re-sense for information that has already been sensed for.

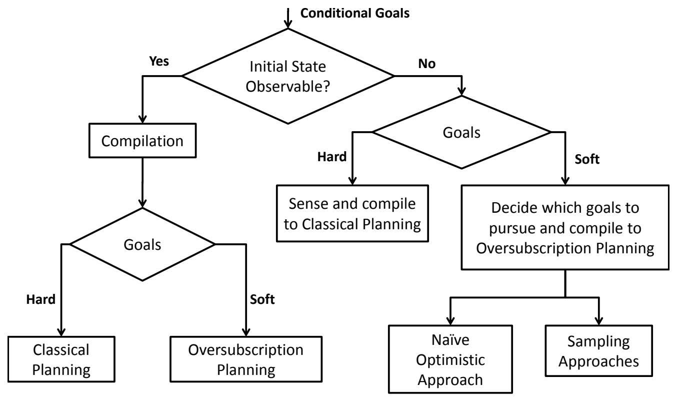

Figure 1: A schematic outline of methods to deal with Conditional Goals.

through the problem update process. The goals generated by $\mathcal{G}$ are similarly added. This process is repeated for every new object that F may instantiate.

We treat $\mathcal{P}$ as an *optimistic closure condition*, meaning a particular state of the world is considered closed once the ground closure condition is true. On every update the ground closure conditions are checked and if true the facts in the corresponding ground values from $\mathcal{C}$ and $\mathcal{G}$ are removed from the problem. By planning over this representation, we provide a plan that is executable given the planning system's current representation of the world until new information can be discovered (via a sensing action returning the closure condition). The idea is that the system is interleaving planning and execution in a manner that moves towards rewarding goals by generating an optimistic view of the true state of the world.

Conditional Goals

Open World Quantified Goals can however really be viewed as a specific instantiation of a more general class of goals, known as "conditional goals".

Conditional Goal Given ground predicates A and B, a *conditional goal* $A \rightsquigarrow B$ is defined as the requirement that if A is true in the initial state I, then any plan ρ is a solution to a given planning problem if and only if B is true in the final state resulting from applying ρ in I.

Note that conditional goals may be labeled as "hard" or "soft" using the same criteria as those used in Partial Satisfaction Planning (PSP): a goal is hard if every plan *must* achieve it in order to succeed, and soft otherwise. In this setting, OWQGs can be seen as an "optimistic" determinization of conditional goals, where we always assume that the condition associated with a given goal is true, and consequently add that goal to the set of goals to be achieved. For example, in the USAR scenario, the partial observability of the world is resolved by optimistically assuming the presence of victims in all known rooms, and all associated goals are included in the set of goals to be achieved.

Conditional goals themselves can be seen as constructs that condition on the observability of the initial state of a given planning problem in order to provide a complete compilation to known methods of achieving a given set of goals. In specific, if the initial state is fully observable, then these goals can be compiled away into simple classical or oversubscription planning (depending on if they are hard or soft respectively) by observing the values of the antecedents of each goal to decide its inclusion in the set of goals to be achieved. When the initial state is partially observable, things get more interesting: if it is known that the conditional goal c under consideration is a hard goal, then there is no option but to sense completely to resolve the uncertainty in the initial state. However, if c is known to be a soft goal, then there is the additional problem of weighing the benefits of achieving c versus the cost of sensing for its antecedent. The problem is further compounded in the presence of multiple soft goals (conditional or otherwise), in which case an "expected net benefit" analysis must be performed. A detailed outline of the various ways of dealing with conditional goals is provided in figure 1. We now illustrate the process of planning for these goals with an example.

Suppose the planner decided to sense the conditional goals $\mathcal{G}_c^i : \{P_1^i \rightsquigarrow G_1^i, P_2^i \rightsquigarrow G_2^i, \ldots, P_k^i \rightsquigarrow G_k^i\}$. We analyze the costs and benefits of this decision. First, let $\mathcal{S}(\mathcal{G}_c^i)$ denote the cost of sensing the status of the conditions $\{P_1^i \cdots P_k^i\}$. The results of sensing cannot be predicted at planning time; to decide whether this sensing cost will be offset by the increased net benefit, the planner has to compute the *expected* net benefit achievable. In order to do this, it needs to have (or assume) some prior knowledge on how the truth values of the antecedents $\mathbf{P} : P_i$ of the conditional goals are jointly distributed. Let this distribution be $\Psi(\mathbf{P})$. Further, let $\mathcal{G}_c^i \setminus \mathbf{P}$ be the set of conditional goals that are triggered by a specific valuation of the antecedents. For each such valuation $\mathbf{P}$, the optimal net benefit achievable by the planner is $\mathcal{B}(G_o \cup [\mathcal{G}_c^i \setminus \mathbf{P}])$. The expected net benefit is $\mathbf{E}_{\mathbf{P}\sim\Psi}\mathcal{B}(G_o \cup [\mathcal{G}_c^i \setminus \mathbf{P}])$. The crux of the problem thus lies in efficiently computing the expectation, and this can be done either by making assumptions on the form of Ψ (e.g. using

OWQGs), or sampling from Ψ. The optimal set of conditional goals to be sensed $\hat{\mathcal{G}}_c$ is computed as:

$$\hat{\mathcal{G}}_c = \arg\max_{\hat{\mathcal{G}}_c^i \subseteq \mathcal{G}_c} \mathbf{E}_{\mathbf{P} \sim \Psi} \mathcal{B}(G_o \cup [\mathcal{G}_c^i \setminus \mathbf{P}]) - \mathcal{S}(\mathcal{G}_c^i)$$

Alternative Formalisms Thus far, we have examined methods of dealing with conditional goals that fall within the purview of deterministic planning. However, when additional information about objects in the world (and facts about those objects) is available, it would be unwise to ignore such knowledge. Very often, such information is specified in probabilistic terms, via distributions (such as Ψ above). There exist a number of established formalisms within the planning literature to handle such scenarios[2]. Partially observable worlds may be handled by POMDPs; unknown objects (and open-worlds in general) may be added by moving to a first-order MDP framework. Sensing actions can be accomodated too, by using belief-state MDPs and casting them as *information-gathering* MDPs. Combining these frameworks into the system required to solve a problem such as the USAR scenario presented previously presents an exciting line of future research.

Expected Impact

The contributions described in this paper may be viewed as steps toward solving the problem of enabling planners (in their current mould) to make plans for and control robots acting in real-world application scenarios. A common criticism that is leveled at application work in the planning community is that it is often too disconnected from the state-of-the-art in academic planning, and thus tends not to affect the evolution of the latter. Existing industrial applications of planning such as interplanetary rovers and modular printers are hard to integrate in an out of the box fashion with the current best planners; indeed, to achieve the levels of robustness that these scenarios demand, either specialized planning algorithms or carefully crafted problems are required. This also explains the paucity of robotic applications that use frontline planners; planning for robots requires a degree of low-level detail that cannot be handled by current planning formalisms without significantly increasing the size of the problem.

The hope for the future of this work is that it will result in methods of capturing and translating useful knowledge available from the application scenario into a format that planning systems may use to generate better plans. An example of such a task would be generating or learning probabilistic distributions over the occurrence of certain problem features in specific domains – in the USAR scenario, for example, this may entail things like specifying and subsequently using distributions over humans being in specific types of rooms. More generally, the logical culmination of such a direction of work would be to automate the creation of benchmarks based on detailed data from the application scenario, thus eliminating the burden of domain modeling and the various errors that accompany it. We believe that this is a very promising direction of work.

[2]The author wishes to thank the anonymous reviewer for pointing out the connections that follow.

Conclusion

In this paper, we presented a novel approach to reconcile a planner's closed world representation with the open world that a robot has to typically execute it. To enable this approach, we presented the integration of techniques that, combined, are sufficient to represent and solve the scenario described. We showed that we could handle information about new objects in the world using open world quantified goals, and that our replanning and execution monitoring system is able to handle the new information specified by these goals in order to produce plans that achieve a higher net benefit. We also discussed that our system could support soft goals, thus ensuring that opportunities retain their bonus nature, and do not turn into additional hard goals that may constrain existing hard goals.

In addition, we showed that open world quantified goals can be seen as a special case of a more general class of goals known as conditional goals, and illustrated the process of computing the expected net benefit of a given set of such goals. We are currently looking into ways of extending this work by using hindsight optimization and anticipatory planning techniques (Yoon et al. 2008; Hubbe et al. 2008). Methods such as these would likely produce a more robust system capable of better balancing sensing costs with expected reward. We are also considering methods of counterfactual domain analysis to determine what objects should be attended to by the robotic system before plan execution begins, in a bid to limit sensing cost and direct the planner better.

References

Cushing, W.; Benton, J.; and Kambhampati, S. 2008. Replanning as a deliberative re-selection of objectives. Technical report, Arizona State University, CSE Department.

Etzioni, O.; Golden, K.; and Weld, D. S. 1997. Sound and efficient closed-world reasoning for planning. *AIJ* 89(1-2):113–148.

Garland, A., and Lesh, N. 2002. Plan evaluation with incomplete action descriptions. In *AAAI 2002*, 461–467.

Hubbe, A.; Ruml, W.; Yoon, S.; Benton, J.; and Do, M. 2008. On-line Anticipatory Planning. In *Workshop on a Reality Check for Planning and Scheduling under Uncertainty, ICAPS 2008.*

Talamadupula, K.; Benton, J.; Schermerhorn, P.; Scheutz, M.; and Kambhampati, S. 2010. Integrating a Closed-World Planner with an Open-World Robot: A Case Study. *To appear: Proceedings of AAAI 2010.*

van den Briel, M.; Sanchez, R.; Do, M.; and Kambhampati, S. 2004. Effective approaches for partial satisfaction (over-subscription) planning. In *AAAI 2004*, 562–569.

Yoon, S.; Fern, A.; Givan, R.; and Kambhampati, S. 2008. Probabilistic planning via determinization in hindsight. In *AAAI 2008*.