Goal-Driven Autonomy Agents with Sensing Costs

Dustin Dannenhauer and **Héctor Muñoz-Avila**
Lehigh University
Bethlehem, PA 18015
{dtd212, hem4}@lehigh.edu

Sravya Kondrakunta
Wright State University
Dayton, OH 45435
kondrakunta.2@wright.edu

Abstract

Goal-driven autonomy (GDA) agents can change their goals as needed. They are designed for dynamic, open, and partially observable environments. In this paper we study the formalization of goal-driven autonomy agents operating in these kinds of environments, with the assumption that agents have the capability to sense the environment with some associated costs. We examine goal-driven autonomy agents that operate in these domains, considering multiple approaches to select which subset of the agent's sensing actions (given an associated cost) to execute. The contributions of this work are the following: (1) a complexity analysis of the problem, (2) improvements over previous goal-driven autonomy to decide which sensing actions to perform and (3) empirical results demonstrating the improvement of these new approaches.

1 Introduction

Autonomous agents face difficulties in environments that are dynamic and partially observable. We look at this problem from the perspective of goal-driven autonomy (GDA) agents, which may decide to change their goals in response to changes in the environment [Muñoz-Avila *et al.*, 2010a; Molineaux *et al.*, 2010]. The actions an agent takes are likely to be different depending on the agent's goals. In environments that are partially observable, the agent may not be able to directly observe all information in the state. Since some of this information is relevant for deciding which goal to pursue, we consider agents endowed with sensing capabilities. Sensing often comes at a cost, and therefore the aim of this work is to minimize sensing cost during the GDA process.

A formal representation of this problem has been described in [Dannenhauer *et al.*, 2016] as the guiding sensing problem. Briefly, this is a formalism of agents described above. We will re-formulate this problem in Section 2 below. Informally, the guiding sensing problem aims to generate solutions that minimize the sensing costs. Unlike POMDPs, we don't assume that the dynamics of the environment are known. Researchers have pointed out that in many domains knowing information about probability distributions of events is unfeasible [Brenner and Nebel, 2006]. Furthermore, the formalization simplifies the goal reasoning process by requiring that at least one goal among a set of goals G is achieved.

Research on goal-driven autonomy combines planning and execution and the capability of the agent to change its goals [Muñoz-Avila *et al.*, 2010b; Molineaux *et al.*, 2012; Weber, 2012]. The end result is a plan (combining planning and sensing actions) that achieves some set of goals. The guiding sensing problem explicitly considers costs (i.e., sensing costs) and thus introduces quality considerations into the GDA research.

An optimal solution to the guiding sensing problem can only be known in hindsight, since the environment's dynamics (e.g., the probability that certain events will take place) are not known and may change as the agent is acting in the environment.

The contributions of this paper are as follows:

- We provide a complexity analysis of the guiding sensing problem, showing it to have a lower bound complexity PSPACE-hard.
- We introduce new approaches for expectations that vary the frequency at which sensing occurs.
- We empirically evaluate these new approaches against previous approaches to sensing in the partially observable and dynamic domain Marsworld.

This paper is organized as follows: in Section 2 we re-formulate the guiding sensing problem (originally described in [Dannenhauer *et al.*, 2016]). Section 3 gives a complexity analysis of the guiding sensing problem. Section 4 discusses new approaches to solving the guiding sensing problem. Section 5 discusses related work including a discussion of our re-formulation of the guiding sensing problem compared to [Dannenhauer *et al.*, 2016]. Section 6 describes experimental evaluations and Section 7 concludes the paper with a discussion of future work.

2 The Guiding Sensing Problem

We re-formulate the guiding sensing problem (GSP), originally stated in [Dannenhauer *et al.*, 2016]. In the GSP formalism, the collection of actions Σ are divided into two disjoint collections: Σ_{plan}, the planning actions, and Σ_{sense}, the sensing actions.

A planning action $a \in \Sigma_{plan}$ consists of the usual triple, $(prec(a), a^+, a^-)$ indicating the preconditions, positive effects and negative effects of a. Planning actions indicate actions in the domain such as moving the agent to a neighboring location from its current location.

Sensing actions are used to sense all conditions that can be satisfied by sensing in the environment. For every condition τ that may be satisfied in the environment, we define a sensing action $\gamma_\tau \in \Sigma_{sense}$. In particular, for every effect e of an action, there is a unique condition τ_e for that effect. For example, if e = "*beacon 55 is activated*", then γ_{τ_e} is the execution of the sensing action to check if beacon 55 is indeed activated. When a sensing action is performed the current partial state is updated accordingly.

Each sensing action γ_τ has an associated cost, $\mathbf{c}(\gamma_\tau)$. The cost sensing function $c : \Sigma_{sense} \rightarrow \mathbb{R}_{\geq 0}$ returns a non-negative number for each sensing action.

We are interested in agents acting in partially observable and dynamic environments. If Q is the collection of all states in the world, then a collection of atoms s is a *partial state* if there is a state $q \in Q$ such that $s \subseteq q$ holds. We denote by $\mathbf{S}$ the set of all partial states; if Q is finite, then S is finite too.

The 4-tuple input to the guiding sensing problem is defined as (Σ, s_0, G, c). $\mathbf{s_0}$ denotes an initial partial state, and $\mathbf{G}$ denotes a collection of goals.

The guiding sensing problem is defined as follows: given a guiding sensing problem (Σ, s_0, G, c), generate a sequence of actions $\pi = (a_1 ... a_m)$, each $a_k \in \Sigma$, and a sequence of partial states $(s_0 ... s_m)$ such that:

1. If $\pi_{plan} = (a_{k1} ... a_{kn})$ denotes the subsequence of all planning actions in π (i.e., each $a_{kj} \in \Sigma_{plan}$), then the preconditions of each a_{kj} were valid in the environment at the moment when a_{kj} was executed.
2. One or more goals $g \in G$ hold in s_m.
3. If $\pi_{sense} = (a_{k1} ... a_{kz})$ denotes the subsequence of all sensing actions in π (i.e., each action in π_{sense} is of the form $\gamma_\tau()$ for some condition τ), then the total sensing cost $C(\pi) = \sum_{i=1}^{n} c(a_{ki})$ is minimal.
4. Sensing actions must occur between each pair of contiguous planning actions a_{ki}, a_{ki+1} in π_{plan}, one for each effect e of a_{ki}.

Condition 1 guarantees that the actions taken while the agent was acting in the environment were sound by checking if an action's preconditions are valid before executing the action.

Condition 2 guarantees that at least one of the goals is achieved. It is a simplification of the goal-reasoning process where agents can change its goals over time and as the circumstances of the environment change. For a formalization of goal reasoning please see [Cox *et al.*, 2017].

Condition 3 represents an ideal condition where the agent minimizes the cost of sensing while achieving its goals. Condition 4 requires that each effect of a planning action committed to the plan is checked in the environment before committing to the next planning action.

Even though action's costs are not explicitly included in the model, as we are going to see in the next section our model subsumes action's costs.

3 Complexity

Although the guiding sensing problem doesn't explicitly consider actions' costs, we are going to show that action costs are subsumed in this formalization. We do by proving that PLANMIN $\leq_p$ PLANSENSE. PLANMIN is the decision problem for generating plans of minimal length and PLANSENSE is the decision problem for the guiding sensing problem. This implies that the guiding sensing problem has a lower bound complexity of PSPACE-hard.

Definition. (PLANMIN) Given $k \geq 1$ and a STRIPS planning problem $\Pi_{plan} = (\Sigma_{plan}, s_0, G)$, is there a solution plan π_{plan} for Π_{plan} such that π_{plan} has at most k steps?

Definition. (PLANSENSE) Given $m \geq 1$ and a guiding sensing problem $\Pi_{sense} = (\Sigma, s_0, G, c)$, is there a solution plan π for Π_{sense} such that $C(\pi) \leq m$?

We define a polynomial-time reduction from PLANMIN to PLANSENSE as follows: given $\Pi_{plan} = (\Sigma_{plan}, s_0, g)$, we construct $\Pi_{sense} = (\Sigma, s_0, G, c)$ as follows:

1. We modify each planning action $a \in \Sigma_{plan}$ by adding a unique effect e^a.
2. For each effect e of each planning action $a \in \Sigma_{plan}$ we define a unique sensing action γ_{τ_e} (this includes the effects added in Step 1). We define the collection of sensing actions as $\Sigma_{sense} = \{\gamma_{\tau_e} |$ *e is an effect of an action* $a \in \Sigma_{plan}\}$.
3. The evaluation of γ_{τ_e} is always satisfied for every action in Σ_{sense}.
4. We define $\Sigma = \Sigma_{plan} \cup \Sigma_{sense}$
5. We define $G = \{g\}$.
6. For each effect e in each action $a \in \Sigma_{plan}$ that was an effect of the action prior to Step 1, we define $c(\gamma_{\tau_e}) = 0$. For each effect e added in Step 1, we define $c(\gamma_{\tau_e}) = 1$

The following steps: (1) the modification in Step 1, (2) the construction of Σ_{sense} in Step 2, and (3) the construction of the cost function c in Step 6 are each linear on the number of actions in Σ_{plan}.

In Step 3 we make each sensing action γ_{τ_e} always satisfied because we are simulating classical planning and, hence, the effects of the actions are always satisfied following the STRIPS assumption [Fikes and Nilsson, 1971].

The additional and unique effect e^a added for each action a in Step 1 and the cost function in Step 6, results in $C(\pi)$ counting the number of planning actions of any plan π solving the guiding sensing problem.

Step 5 guarantees that g will be satisfied since Condition 2 of the guiding sensing problem requires that at least one goal in G is satisfied. As a result, PLANMIN has a solution plan of length at most k if and only if PLANSENSE has a solution plan of cost at most m = k. Since the transformation from $\Pi_{plan} = (\Sigma_{plan}, s_0, g)$ to $\Pi_{sense} = (\Sigma, s_0, G, c)$ is polynomial, we conclude that $\Pi_{plan} \leq_p \Pi_{sense}$.

4 Computing Agent's Expectations

GDA agents monitor their expectations against the observed state checking for discrepancies. There are multiple ways to compute an agent's expectations (for a formal description see [Dannenhauer *et al.*, 2016]):

- **No expectations.** This is intended as a baseline. The agent doesn't check the effects of the actions executed in the environment.
- **Immediate expectations.** Follows directly from Condition 4 of the guiding sensing problem: the agent checks if the effects of each action are valid in the environment.
- **Informed expectations.** Cumulates the effects of the actions performed so far. It takes into account that some effects might be deleted by subsequent actions already executed and hence the agent does not need to check for those.
- **State expectations.** The agent checks each of the conditions the state.

We now introduce new ways to compute expectations which improve upon informed expectations. A primary benefit of informed expectations is that it can be used for policy planners in which an agent decides what action to take based on the current state (as opposed to generating a single sequential non-branching plan beforehand). An underlying assumption of informed expectations is that each action the agent takes is relevant to later actions and/or the agent's goal.

Informed expectations guarantee that the agent actually achieves its goal when it believes it achieves its goal. Prior experiments from [Dannenhauer *et al.*, 2016] regarding informed expectations measured goal achievement, in these new approaches presented here, we enable agents to perform additional sensing in situations where the agent incorrectly believes it has achieved its goal (which can happen due a discrepancy between the true state and the agent's state). This allows agents to verify the conditions of their goal upon believing they have reached their goal. Given new observations, if the goal is not achieved, the agent continues acting. Such agents can vary the frequency at which they perform sensing, since they will continue acting following an incorrect assumption that they have reached their goal. The choice of which facts of the state should be verified through sensing remain the same as those computed by the original informed expectations.

Frequency refers to how often sensing should be performed whereas expectations refer to what should be sensed. When the number of expectations to check are numerous (such as informed and state expectations), it is unlikely most of these expectations will be violated at once. Therefore, sensing may be able to occur less frequently, without significant hindrance on performance. In this work, a frequency $f = 1$ signifies the agent will perform sensing of informed expectations following each plan action. A frequency $f = 2$ signifies the agent will perform sensing of informed expectations every 2 actions, and so forth for $f = 2, 5, 10, 20$. Whenever $f > 1$ holds, for every step that the agent is not performing sensing of the informed expectations, it will still check immediate expectations to ensure each action is executed successfully.

The algorithm for a goal-driven autonomy agent using the different kinds of expectations is given in [Dannenhauer *et al.*, 2016]. That algorithm includes a function for computing expectations $X(s, \pi)$ and gives five implementations of this function (including immediate, informed and state as informally defined in this section). We add the following implementations:

X_2: informed expectations, frequency = every 2 actions
X_5: informed expectations, frequency = every 5 actions
X_{10}: informed expectations, frequency = every 10 actions
X_{20}: informed expectations, frequency = every 20 actions
X_∞: informed expectations only to be checked at time of believed goal achievement

Whenever the system changes to pursue a goal g, if g has not been tried before, the agent will reset the informed expectations. The agent will start accumulating expectations from the first action achieving g. If g has been tried before and a sequence of actions $\pi_g = a_1, a_2, ..., a_n$ had been executed when pursuing goal g, informed expectations will be computed over π_g. For example, suppose the agent begins pursuing some goal g_1 and takes actions a_1, a_2, a_3 then switches to some other goal g_2 and takes actions a_4, a_5. The expectations while the agent is pursuing goal g_1 will be computed from a_1, a_2, a_3 while the expectations for pursuing goal g_2 will be computed from a_4, a_5. If the agent switches back to g_1 and executes an action a_6 then informed expectations are computed over a_1, a_2, a_3, a_6.

5 Experimental Evaluation and Results

We implemented the new expectations X_i in the Marsworld domain from [Dannenhauer *et al.*, 2016]. Marsworld is a dynamic and partially observable simulated environment that contains randomly located resources for an autonomous agent to use in its pursuit of its goals. Marsworld-like domains have been used in goal-reasoning literature before (see Mudworld from [Molineaux and Aha, 2014] and a slightly different variation of Marsworld from [Dannenhauer and Muñoz-Avila, 2015]). The high-level task of the agent is to make a signal. A signal can be made by activating enough resources. Here, goals are states that contain a minimum number of activated objects (i.e. beacons are activated by being turned on, wood piles are activated by lighting them on fire, and flares are activated if they are lit). So a goal requiring x resources will be to have any x number of beacons activated, x number of flares lit, or x number of wood pile fires. As the agent explores the environment, flares, beacons, and fires may become deactivated (fires and flares become extinguished by wind, beacons may fail on their own). When fires or flares become extinguished, they are no longer usable; beacons can be re-activated if they were previously deactivated.

Here Marsworld is a 10 by 10 tiled grid populated with randomly placed beacons and wood piles (25 each) and the agent starts with an inventory of 25 flares. A single tile will never have more than one resource object, (an agent cannot drop a flare when there is a beacon activated). There is a 35% chance per action executed that a single beacon may become deactivated if it is not already deactivated. Additionally, fires and flares each have an independent chance (also 35%) of failure for a beacon, flare, or fire per action executed providing a high level of dynamism. Each sensing action has a cost of 1.

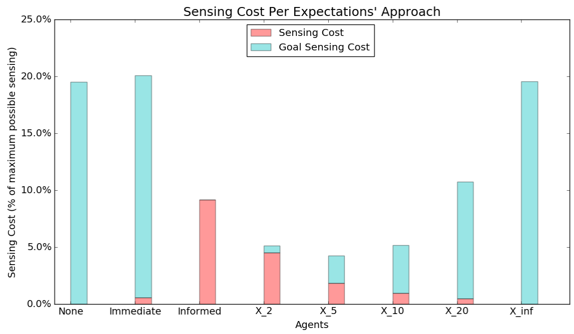

Figure 1: Sensing Costs per Approach

Figure 1 shows the cumulative results of an agent over 100 randomly generated scenarios. Each agent achieves all of its goals. The x-axis is the type of expectations that agent is using (all agents are identical except for expectations). The y-axis shows the percentage of maximum possible sensing that was performed. Bars are divided into two colors, red and blue, with red signifying the amount of sensing that was done prior to the agent believing it reached its goal, and with blue signifying the amount of sensing the agent performed to verify its goal conditions were met. Goal sensing is only performed when the agent believes it reaches its goal but hasn't actually (hence informed does not require any goal sensing costs).

The first two bars show the performance of agents using None and Immediate expectations. Their behavior is as we expected: no or minimal sensing is done prior to goal achievement, and as a result the agents spend most or all of their sensing costs checking to see their goal is actually achieved. The third bar, which is the original informed expectations guarantees that the goal is always reached when the agent believes it is reached, and this reduces sensing from immediate. We hypothesized that at least some of the new approaches to expectations (X_2, X_5, X_{10}, X_{20}) will reduce sensing and we observe this to be true for X_2, X_5, and X_{10}. Agent using a frequency of 20 (X_{20}) is close to the original informed expectations and X_5 performs best. These results show that varying the frequency of sensing can reduce the overall sensing cost, but the rate of sensing is important.

6 Related Work

The guiding sensing problem was originally formulated in [Dannenhauer *et al.*, 2016]. In addition to the input (Σ, s_0, G, c) as defined here, it included two more elements: the partial states S that the agent could visit and a heuristic function $\phi_g : S \rightarrow \mathcal{A}$ that tells the agent which action a to perform when it finds itself in partial state s and pursuing goal g. We feel that explicitly requiring the partial states and control heuristic was unnecessary just like when defining the STRIPS planning problem we do not need to provide the collection of states or make any commitment of how plans are generated. Additionally, in our new definition we add Condition 4 requiring that the effects of the action executed are checked with sensing actions. This was needed because the original definition did not make any commitments about when the sensing actions would be executed.

The problem of planning in dynamic environments spawned contingency planning methods [Dearden *et al.*, 2003] in which agents plan for plausible events and conditions that may occur during plan execution. Conformant planning methods [Goldman and Boddy, 1996] generate plans that are guaranteed to succeed given some strong assumptions such as the a priori identification of all possible contingencies. Plan repair methods instead adapt a plan's remaining actions whenever the state conditions required to execute the plan's next action are not satisfied [Fox *et al.*, 2006]. These agents cannot change their goals, whereas GDA agents dynamically reason about which goals they should achieve or modify.

Deterministic (STRIPS) planning assumes that actions have a predetermined outcome [Fikes and Nilsson, 1971]. The result of planning is a sequence of actions that enable the agent to achieve its goals. A Markov Decision Process (MDP) is a frequently studied planning paradigm whereby actions have multiple outcomes [Howard, 1960]. In MDPs, solutions are found by iterating over the possible outcomes until a policy is generated which indicates for every state that the agent might encounter, what action to take that will enable the agent to achieve its goals. A Partial Observable Markov Decision Process (POMDP) is an extension of MDP for planning when the states are partially observable [Kaelbling *et al.*, 1998]. In POMDPs, solutions are found by iterating over the possible states that the agent believes itself to be in and the possible outcomes of the actions taken on those states until a policy is found. The GDA framework is general allowing a variety of planning paradigms to be adopted as the planner Π. GDA research has used both planning [Molineaux *et al.*, 2010] and MDP-based planning [Jaidee *et al.*, 2012]. Also in regard to POMDPs, the goal-sensing problem doesn't assume that the dynamics of the environment are known by the agent.

The general topic of combining planning and execution has, of course, a long history [Goldman *et al.*, 1996]. For example, Sage will aim to plan as far as possible with the known information and perform sensing when needed to advance the plan further [Knoblock, 1995]. There is a recurrent interest on planning and execution as exemplified by the recent call for the actor view of planning [Ghallab *et al.*, 2004]. Brenner and Nebel coined the term continual planning to refer to the integration of planning, execution and monitoring [Brenner and Nebel, 2006]. In their work sensing actions are defined by using variables that are allowed to be uninstantiated. So for example, the result of a sensing action changes the status of a variable from undefined to a particular constant. An algorithm for asynchronous planning and execution monitoring is presented. Bonet and Geffner [Bonet and Geffner, 2014] study the problem of contingent planning (i.e., generation of tree plan that accounts for all contingencies that might occur during execution) and conformant planning (i.e., plans that are guaranteed to succeed regardless of the uncertainty in the environment) in belief states (i.e., the collection of all states that are consistent with the current set of observations). Conformant [Goldman and Boddy, 1996] and contingency [Pryor and Collins, 1996] planning are particularly useful in situations when the probability distributions are not known and hence fall outside of the POMDP framework. Bonet and Geffner's formalism use multi-valued variables and conditional effects to model uncertainty in the environment. Their results show that belief tracking (i.e., planning with belief states) is Turing-complete and propose an approximation algorithm using factored representations.

An alternative to contingent and conformant planning in dynamic environments is replanning [Shani and Brafman, 2011]. A plan is generated and when an execution failure is encountered, a new plan is generated from the state where the failure occurred. This has been extended for planning in belief states [Shani and Brafman, 2011]. The GDA framework could adopt any of these planning paradigms for the planning phase. In our work, our planning phase is reminiscent of replanning. The crucial characteristic of GDA is that the agent can change its goals over time.

The representation of goals in partially observable environments that require exploration may need to be different than in fully observable environments. [Talamadupula *et al.*, 2010] introduce the concept of Open World Quantifiable Goals used in an urban search and rescue setting. In that work, human-robot teams search for survivors in damaged buildings, and because the number of survivors is not known ahead of time, they use goal structures that award an agent a higher score for rescuing more survivors while balancing a goal to survey an area in a limited amount of time. For future work we would like to explore the compatibility of Open World Quantifiable Goals into these kinds of goal reasoning agents.

The goal selection operation can be performed in several ways. For example, in the ICARUS architecture [Choi, 2011] goal selection is based on the priority values assigned to the goals, the values assigned are in the range 0 to 10, 0 signifies that the goal has the least possible priority and 10 indicates highest priority. In this work, we use a heuristic that chooses the goal we are closest to achieving and do not consider priorities (as the focus of this work is on expectations).

7 Conclusion and Future Work

Dynamic and partially observable environments present a challenge for agents with sensing capabilities: how to maximize goal achievement while reducing sensing. Optimal sensing can only be known in hindsight, a perfect solution would involve an agent "magically" knowing what fact outside its view will change and then sense it immediately following that change. The solutions given in this paper improve upon previous approaches for guiding sensing and reducing overall sensing cost.

The specific contributions of this paper are: (1) a reformulation of the guiding sensing problem followed by a complexity proof that the guiding sensing problem has a lower-bound complexity of PSPACE-hard in Section 3, (2) new approaches that vary the frequency of sensing, in order to reduce overall sensing while still achieving goals, and (3) empirical results showing the benefit from sensing.

While an optimal solution to minimal sensing is unavailable, it is our opinion that even further improvements can be made from the new approaches described in Section 4. This leaves multiple areas for future work, including:

- The results from Section 5 show that varying the frequency of sensing leads to reducing overall sensing costs. However finding the best frequency rate is important. An agent with too sparse a frequency rate (compared to how many actions it executes), may incur higher overall sensing because it fails to achieve its goal too many times, leading to more overall sensing (as is the case with $X_1 0$ and $X_2 0$ in Figure 1). A future approach could be to use a reinforcement-learning like technique to decide to sense a particular condition of a state with some probability correlated to how much time has passed since the agent last sensed that condition.
- We would like to examine the relationship between the rate of change (dynamism) of the environment and the ideal frequency of sensing. We hypothesize that the more dynamic the environment, the more important that the frequency is smaller.
- We would like to consider a model of sensing costs that is non-uniform, such that some sensing actions have a higher cost than others (i.e. the farther away an object is from the agent, the more expensive it is to sense). Perhaps the cost of a particular action should be considered in deciding whether to perform that sensing action in order to minimize overall sensing cost.

Acknowledgements: This work was supported in part by grants ONR N00014-15-1-2080 and NSF 1217888.

References

[Bonet and Geffner, 2014] Blai Bonet and Hector Geffner. Belief tracking for planning with sensing: Width, complexity and approximations. *Journal of Artificial Intelligence Research*, 50:923–970, 2014.

[Brenner and Nebel, 2006] Michael Brenner and Bernhard Nebel. Continual planning and acting in dynamic multiagent environments. In *Proceedings of the 2006 international symposium on Practical cognitive agents and robots*, pages 15–26. ACM, 2006.

[Choi, 2011] Dongkyu Choi. Reactive goal management in a cognitive architecture. *Cognitive Systems Research*, 12(3):293–308, 2011.

[Cox *et al.*, 2017] Michael T Cox, Dustin Dannenhauer, and Sravya Kondrakunta. Goal operations for cognitive systems. In *AAAI*, pages 4385–4391, 2017.

[Dannenhauer and Muñoz-Avila, 2015] D. Dannenhauer and H. Muñoz-Avila. Raising Expectations in GDA Agents Acting in Dynamic Environments. In *International Joint Conference on Artificial Intelligence (IJCAI-15)*, 2015.

[Dannenhauer *et al.*, 2016] D. Dannenhauer, H. Munoz-Avila, and M. T Cox. Informed Expectations to Guide GDA Agents in Partially Observable Environments. In *International Joint Conference on Artificial Intelligence (IJCAI-16)*, 2016.

[Dearden *et al.*, 2003] Richard Dearden, Nicolas Meuleau, Sailesh Ramakrishnan, David E Smith, and Rich Washington. Incremental contingency planning. 2003.

[Fikes and Nilsson, 1971] Richard E Fikes and Nils J Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208, 1971.

[Fox *et al.*, 2006] Maria Fox, Alfonso Gerevini, Derek Long, and Ivan Serina. Plan stability: Replanning versus plan repair. In *ICAPS*, volume 6, pages 212–221, 2006.

[Ghallab *et al.*, 2004] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated planning: theory & practice*. Elsevier, 2004.

[Goldman and Boddy, 1996] Robert P Goldman and Mark S Boddy. Expressive planning and explicit knowledge. In *AIPS*, volume 96, pages 110–117, 1996.

[Goldman *et al.*, 1996] R Goldman, M Boddy, and Louise Pryor. Planning with observations and knowledge. In *AAAI-97 workshop on theories of action, planning and control*, 1996.

[Howard, 1960] Ronald A Howard. Dynamic programming and markov processes.. 1960.

[Hunter, 2007] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[Jaidee *et al.*, 2012] Ulit Jaidee, Héctor Muñoz-Avila, and David W Aha. Learning and Reusing Goal-Specific Policies for Goal-Driven Autonomy. In *Case-Based Reasoning Research and Development*, pages 182–195. Springer, 2012.

[Kaelbling *et al.*, 1998] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

[Knoblock, 1995] Craig A Knoblock. Planning, executing, sensing, and replanning for information gathering. In *In Proceedings Of The Fourteenth International Joint Conference On Artificial Intelligence*, 1995.

[Molineaux and Aha, 2014] Matthew Molineaux and David W Aha. Learning Unknown Event Models. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[Molineaux *et al.*, 2010] Matt Molineaux, Matthew Klenk, and David W Aha. Goal-driven autonomy in a navy strategy simulation. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 1548–1554. AAAI Press, 2010.

[Molineaux *et al.*, 2012] Matthew Molineaux, Ugur Kuter, and Matthew Klenk. Discoverhistory: Understanding the past in planning and execution. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 989–996, 2012.

[Muñoz-Avila *et al.*, 2010a] Héctor Muñoz-Avila, David W Aha, Ulit Jaidee, Matthew Klenk, and Matthew Molineaux. Applying goal-driven autonomy to a team shooter game. In *FLAIRS Conference*, 2010.

[Muñoz-Avila *et al.*, 2010b] Héctor Muñoz-Avila, Ulit Jaidee, David W Aha, and Elizabeth Carter. Goal-Driven Autonomy with Case-Based Reasoning. In *Case-Based Reasoning. Research and Development*, pages 228–241. Springer, 2010.

[Pryor and Collins, 1996] Louise Pryor and Gregg Collins. Planning for contingencies: A decision-based approach. *Journal of Artificial Intelligence Research*, 4:287–339, 1996.

[Shani and Brafman, 2011] Guy Shani and Ronen I Brafman. Replanning in domains with partial information and sensing actions. In *IJCAI*, volume 2011, pages 2021–2026, 2011.

[Talamadupula *et al.*, 2010] Kartik Talamadupula, J Benton, Subbarao Kambhampati, Paul Schermerhorn, and Matthias Scheutz. Planning for human-robot teaming in open worlds. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 1(2):14, 2010.

[Weber, 2012] Ben Weber. *Integrating Learning in a Multi-Scale Agent*. PhD thesis, University of California, Santa Cruz, June 2012.