Planning in Stochastic Environments with Goal Uncertainty

Sandhya Saisubramanian[1] and **Kyle Hollins Wray**[1,2] and **Luis Pineda**[1,3] and **Shlomo Zilberstein**[1]
[1]College of Information and Computer Sciences, University of Massachusetts Amherst, Massachusetts, USA
[2]Alliance Innovation Lab Silicon Valley, Santa Clara, California, USA
[3]Facebook AI Research, Montreal, Quebec, Canada

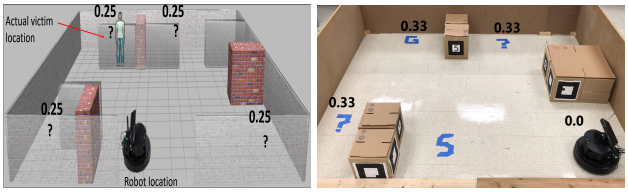
Abstract

We present the Goal Uncertain Stochastic Shortest Path (GUSSP) problem—a general framework to model path planning and decision making in stochastic environments with goal uncertainty. The framework extends the stochastic shortest path (SSP) model to dynamic environments in which it is impossible to determine the exact goal states ahead of plan execution. GUSSPs introduce flexibility in goal specification by allowing a belief over possible goal configurations. The unique observations at potential goals helps the agent identify the true goal during plan execution. The partial observability is restricted to goals, facilitating the reduction to an SSP with a modified state space. We formally define a GUSSP, discuss its theoretical properties, and propose an admissible heuristic that reduces the planning time using FLARES—a start-of-the-art probabilistic planner. We also propose a determinization approach for solving this class of problems. Finally, we present empirical results on a search and rescue mobile robot and three other problem domains in simulation.

Introduction and Related Work

Autonomous robots acting in the real world are often faced with tasks that require path planning in stochastic environments. These problems are typically modeled as a Stochastic Shortest Path (SSP) problem, which generalizes both finite and infinite-horizon Markov decision processes (MDPs) and is a convenient framework to model goal-driven problems (Bertsekas and Tsitsiklis 1991). The objective in an SSP is to devise a sequence of actions such that the expected cost of reaching a *known* goal state from the start state is minimized.

Consider a search and rescue domain (Figure 1), a motivating example where the robot has to devise a cost minimizing path to rescue people from a building (Kitano et al. 1999; Pineda et al. 2015). While the number of victims and the map of the building may be provided to the robot, only potential victim locations may be known ahead of plan execution. The unavailability of the exact goal states (victim locations) during planning time prevents the problem from being modeled as a standard MDP or SSP. In this work we assume that the exact goal states may be hard to identify, but historical data or noisy sensors allow the robot to establish a belief distribution over possible victim locations. The search and rescue domain is an instance of the *optimal search for stationary targets* (Hansen 2007; Stone, Royset, and Washburn 2016; Bourgault, Furukawa, and Durrant-Whyte 2003; Trevizan and Veloso 2013)—a class of problems in which the target's exact location is unknown to the robot, but the robot can observe its current location and determine whether the target is in the current location. Hence, we assume that the robot is given well-defined goal conditions, but has uncertainty about the states that satisfy these goal conditions.



(a) Problem setting (b) Experimental setting

Figure 1: An illustrative example of a search and rescue problem with goal uncertainty, showing a motivating problem setting with the initial belief (left) and the corresponding experimental setting of the problem with a mobile robot and updated beliefs (right). The question marks indicate potential victim locations and values denote the robot's belief. S denotes the robot's start location and G is the actual victim location (goal). The robot updates its belief about the victim locations based on its observations.

In the existing literature (Nie, Wong, and Kaelbling 2016; Ong et al. 2010), such problems are typically modeled as a Partially Observable MDP (POMDP) (Kaelbling, Littman, and Cassandra 1998), a rich framework that facilitates modeling various forms of partial observability. However, POMDPs are much harder to solve (Papadimitriou and Tsitsiklis 1987) than MDPs. The partially observable SSPs (POSSPs) extend the SSP framework to settings with partially observable states, offering a class of indefinite-horizon, undiscounted POMDPs that rely on state-based termination (Patek 2001). Other relevant POMDP variants are the Mixed Observable MDPs (Ong et al. 2010) that model problems with both fully observable and partially observable state factors and the Goal POMDPs (Bonet and Geffner 2009) that are goal-based with no discounting. These models are solved using POMDP solvers and are difficult to solve optimally. They also suffer from limited scalability due to

their computational complexity (Papadimitriou and Tsitsiklis 1987). Extensions of POMDPs to multi-agent problems include goal-directed DEC-POMDPs (Amato and Zilberstein 2009), which extend goal-directed planning to multi-agent settings with partial observability.

Another related line of work is the transition-uncertain MDPs (Delgado et al. 2011) which can capture the uncertainty in transitioning to the goal states. However, solving MDPs with imprecise transitions is complex and designing efficient solvers for this class of problems remains under-explored. Our objective in this work is to develop efficient planners for problems with goal uncertainty by leveraging the fully observable components of the problem.

We present goal uncertain SSP (GUSSP), a framework specifically designed to model problems with imperfect goal information by allowing for a probabilistic distribution over possible goals. GUSSPs fit well with many real-world settings where it is easier and more realistic to have belief over goal configurations, rather than exact knowledge about the goal states. The observation function in a GUSSP facilitates the reduction to an SSP, enabling the computation of tractable and optimal solutions. We address settings where the goals do not change over time and we assume the existence of a unique observation that allows the robot to accurately identify a goal when it reaches one. We define the property of an *order-k* policy that helps understand the complexity of policy execution. This measure bounds the maximum number of unique visits to states that provide information about the goal, before the agent discovers a true goal.

Our key contributions are: (i) a formal definition of GUSSP and its theoretical properties; (ii) a domain-independent, admissible heuristic that can accelerate probabilistic planners; (iii) a determinization approach for solving GUSSPs; and (iv) empirical evaluation on three realistic domains in simulation and on a mobile robot.

Background: Stochastic Shortest Path

A **Stochastic Shortest Path** (SSP) is a more general formulation of an MDP to model goal-oriented problems that require sequential decision making under uncertainty. Formally, an SSP is defined by the tuple $\langle S, A, T, C, s_0, S_G\rangle$, where S is a finite set of states; A is a finite set of actions; $T : S \times A \times S \rightarrow [0, 1]$ is the transition function representing the probability of reaching a state $s' \in S$ by executing an action $a \in A$ in state $s \in S$, and denoted by $T(s, a, s')$; $C : S \times A \rightarrow \mathbb{R}^+ \cup \{0\}$ is the cost function representing the cost of executing action $a \in A$ in state $s \in S$, and denoted by $C(s, a)$; $s_0 \in S$ is the initial state; and $S_G \subseteq S$ is the set of absorbing goal states. The cost of an action is positive in all states except absorbing goal states, where it is zero. An SSP is an MDP with no discounting, that is, the discount factor $\gamma = 1$. The objective in an SSP is to minimize the expected cost of reaching a goal state from the start state. It is assumed that there exists at least one *proper policy*, one that reaches a goal state from any state s with probability 1. The optimal policy, π^*, can be extracted using the value function

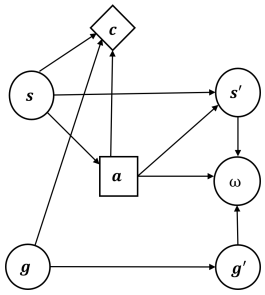


Figure 2: A dynamic Bayesian network describing a GUSSP.

defined over the states, $V^*(s)$:

$$V^*(s) = \min_a \; Q^*(s, a), \quad \forall s \in S$$

$$Q^*(s, a) = C(s, a) + \sum_{s'} T(s, a, s')V^*(s'), \forall (s, a)$$

with $Q^*(s, a)$ denoting the optimal Q-value of the action a in state s in the SSP. While SSPs can be solved in polynomial time in the number of states, many problems of interest have a state-space whose size is exponential in the number of variables describing the problem (Littman 1997). This complexity has led to the use of various approximate methods that either ignore stochasticity or use a short-sighted labeling approach for quickly solving the problem.

Goal Uncertain Stochastic Shortest Path

A goal uncertain stochastic shortest path (GUSSP) problem is a generalized framework to model problems with goal uncertainty. A GUSSP is an SSP in which the agent may not know initially the exact set of goal states (S_G, which does not change over time), and instead can obtain information about the goals via observations.

Definition 1. *A **goal uncertain stochastic shortest path problem** is a tuple $\langle X, S, A, T, C, s_0, S_G, P_G, \Omega, O\rangle$ where*

- *S, A, T, C, s_0, S_G denote an underlying SSP with S_G unknown to the agent;*
- *$P_G \subseteq S$ is the set of potential goals such that $S_G \subseteq P_G$;*
- *$X = S \times G$ is the set of states in the GUSSP with $G = 2^{P_G} \backslash \{\emptyset\}$ denoting the set of possible goal configurations;*
- *Ω is a finite set of observations corresponding to the goal configurations, $\Omega = G$; and*
- *$O : A \times X \times \Omega \rightarrow [0, 1]$ is the observation function denoting the probability of receiving an observation, $\omega \in \Omega$, given action $a \in A$ led to state x' with probability $O(a, x', \omega) \equiv Pr(\omega|a, x')$.*

Each state is represented by $\langle s, g\rangle$, with $s \in S$ and $g \in G$. GUSSPs have mixed observable state components as s is fully observable. Each $g \in G$ represents a goal configuration (set of states), thus permitting multiple true goals in the model, $|S_G| \geq 1$. Every action in each state produces an observation, $\omega \in \Omega$, which is a goal configuration that provides information about the true goals. The agent's belief about its current state is denoted by $b(x)$, with $x = \langle s, g\rangle$; that

is, the belief about $g = S_G$. The initial belief is denoted by $b_0\langle s_0, g\rangle \in [0,1], \forall g \in G$, where s_0 is the start state. SSPs are therefore a special type of GUSSPs with a collapsed initial belief over the goals. The process terminates when the agent reaches a state x such that $b(x)=1$ and $s \in g$. Figure 2 shows a part of the network representation for a GUSSP.

As with (PO)SSPs, we assume that in a GUSSP: (1) there exists a proper policy with a finite cost, (2) all improper policies have infinite cost, and (3) termination is perfectly recognized.

Observation Function In a GUSSP, an observation function is characterized by two properties. First, to perfectly recognize termination, all potential goals are characterized by a unique belief-collapsing (when the belief over a state is either 1 or 0) observation. That is, at potential goal states, if $s' \in g'$, then $\forall a \in A$:

$$O(a, x', \omega) = \begin{cases} 1 & \text{if } g' = \omega \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Second, the observation function is *myopic*, providing information only about the current state. This is based on real-world settings with limited range sensors and the exploration and navigation approaches for robots that acknowledge the perceptual limitations of robots (Biswas and Veloso 2013). Therefore, the nonpotential goal states provide no information about the true goals, $O(a, x', \omega) = \frac{1}{|\Omega|}$. The *landmark states* are special nonpotential goal states that provide accurate information about certain potential goals. Each $s \in L_s$ provides observations about a subset of potential goals with Ω_s denoting the corresponding set of observations. Therefore, the observation function at nonpotential goal states is, $\forall a \in A$:

$$O(a, x', \omega) = \begin{cases} 1 & \text{if } s' \in L_s \wedge \omega \subseteq g' \wedge \omega \in \Omega_{s'} \\ 0 & \text{if } s' \in L_s \wedge \omega \not\subseteq g' \wedge \omega \in \Omega_{s'} \end{cases}, \quad (2)$$

with $x = \langle s, g\rangle$ and $x' = \langle s', g'\rangle$. The potential goals along with the landmark states are called **informative states**, $\mathcal{I} = P_G \cup L_s$, since they provide information about the true goals through deterministic observations. Thus, our observation function satisfies the minimum information required for state-based termination. In the next section, we discuss a more general setting where every state may have a noisy observation regarding the true goals.

Belief Update A belief b is a probability distribution over X, $b(x) \in [0,1], \forall x \in X$ and $\sum_{x \in X} b(x) = 1$. The set of all reachable beliefs forms the belief space $B \subseteq \Delta^n$, where Δ^n is the standard $(n-1)$-simplex. The agent updates the belief $b' \in B$, given the action $a \in A$, an observation $\omega \in \Omega$, and the current belief b. Using the multiplication rule, the updated belief for $x' = \langle s', g'\rangle$ is:

$$\begin{aligned} b'(x'|b,a,\omega) &= Pr(g'|b,a,\omega,s')\, Pr(s'|b,a,\omega,s) \\ &= Pr(g'|b,a,\omega,s')\, T(s,a,s') \\ Pr(g'|b,a,\omega,s') &= \eta Pr(\omega|b,a,s',g') Pr(g'|b,a,s') \\ &= \eta O(a,x',\omega) \sum_{g \in G} Pr(g', g|b,a,s') \\ &= \eta O(a,x',\omega) Pr(g|b,a,s') \\ &= \eta O(a,x',\omega) b(g), \end{aligned} \quad (3)$$

with $\eta = Pr(\omega|b,a,s')^{-1}$ is a normalization constant and $b(g)$ is the belief over the goal configuration. Therefore,

$$b'(x'|b,a,\omega) = \eta O(a,x',\omega) b(g) T(s,a,s'). \quad (4)$$

The above equation reflects that the belief is only over the goal configurations.

Policy and Value The agent's objective in a GUSSP is to minimize the expected cost of reaching a goal, $\min_{\pi \in \Pi} \mathbb{E}\Big[\sum_{t=0}^{h} C(x_t, a_t) \Big| \pi\Big]$, where x_t and a_t denote the agent's state and action at time t respectively, and $h \in \mathbb{N}$ denotes the horizon. A policy $\pi : B \rightarrow A$ is a mapping from belief $b \in B$ to an action $a \in A$. The value function for a belief, $V : B \rightarrow \mathbb{R}$ is the expected cost for a fixed policy π and a horizon h. The Bellman optimality equation for GUSSPs follows from POMDPs:

$$V(b) = \min_{a \in A} \big[C(b,a) + \sum_{\omega \in \Omega} Pr(\omega|b,a) V(b'_{a\omega})\big],$$

where $b'_{a\omega}$ is the updated belief following Equation 4, $C(b,a) = \sum_x b(x) C(x,a)$, $x = \langle s, g\rangle$, and $x = \langle s', g'\rangle$. A proper policy, π, in a GUSSP guarantees termination in a finite expected number of steps, $V^\pi(b_0) < \infty$.

The number of potential goals with non-zero belief values indicate the degree of uncertainty over goals. The problem setting and the optimal policy determine when the belief values collapse to the true goals. When deploying robots in real-world settings with goal uncertainty, it is useful to understand the problem complexity for policy execution. We measure this by the maximum number of unique visits to informative states that may be required before a true goal is discovered by the agent. We consider unique visits since no new information is obtained thereafter. For example, consider a search and rescue domain in which the agent searches for victims in a corridor with the start state on one end and followed by a series of potential goals. If the first potential goal location is a true goal, then the agent visits only one potential goal before the true goal is discovered, following the optimal policy. This property is beneficial especially in environments with landmark states that reveal the true goals, thus minimizing the need to visit the potential goals specifically to determine the true goals.

Definition 2. *A GUSSP policy π is of **order-k** if there are at most k unique visits to informative states before a true goal is reached following π.*

For state-based termination, $1 \leq k \leq |P_G|$. We illustrate this property in our experiments on a robot, using optimal policies corresponding to different initial beliefs.

Theoretical Analysis

In a GUSSP, the observation function critically affects the number of reachable beliefs. We begin with analyzing how the number of beliefs may grow in the more general (non-myopic observation) setting and then show that a GUSSP with myopic observations has finite reachable beliefs.

In a GUSSP with non-myopic observations, the nonpotential goal states provide stochastic observations about the true goals, resulting in infinitely many reachable beliefs. While this is a trivial fact, it is useful to understand the growth in complexity of the problem and it provides an important link to POMDPs via the belief MDP. The following proposition formally proves this complexity.

Proposition 1. *For all horizon $h > 0$, the belief-MDP of a GUSSP with non-myopic observations may have $\mathcal{O}(|\Omega|^h)$ states.*

Proof. By construction, we map this GUSSP to a belief MDP $\langle B, \mathcal{A}, \tau, \rho \rangle$ with a horizon h (Kaelbling, Littman, and Cassandra 1998). Let $\mathcal{R}(b_0)$ denote the set of reachable beliefs in the GUSSP. The set of states in the MDP is the set of reachable beliefs from b_0 in the GUSSP, $B = \mathcal{R}(b_0)$. The set of actions in the GUSSP are retained in the MDP, $\mathcal{A} = A$. The cost function $\rho(b, a) = \sum_{x \in X} b(x)C(x, a)$, where $C(x, a)$ corresponds to cost function of GUSSP. The transition function for the belief MDP is the probability of executing action $a \in \mathcal{A}$ in belief state $b \in B$ and reaching the reaching belief b', and denoted by $\tau(b, a, b')$, is:

$$\begin{aligned}\tau(b, a, b') &= \sum_{\omega \in \Omega} Pr(b', \omega | b, a) \\ &= \sum_{\omega \in \Omega} Pr(b' | b, a, \omega) Pr(\omega | b, a) \\ &= \sum_{\omega \in \Omega} Pr(\omega | b, a)[b' = b'_{a\omega}],\end{aligned}$$

with Iversen bracket $[\cdot]$ and $b'_{a\omega}$ denoting the updated belief calculated using Equation 4, after executing action a and receiving observation ω. The probability of receiving ω is:

$$\begin{aligned}Pr(\omega | b, a) &= \sum_{x' \in X} Pr(\omega, x' | b, a) \\ &= \sum_{x' \in X} O(a, x', \omega) \sum_{x \in X} T(s, a, s') b(g'),\end{aligned}$$

with $x = \langle s, g \rangle$ and $x' = \langle s', g' \rangle$. Since $|S|$ in the GUSSP is finite, a finite set of reachable beliefs in the GUSSP results in a finite set of reachable states in the belief MDP. This is a tree of depth h with internal nodes for decisions and transitions, the branching factor is $\mathcal{O}(|\Omega|)$ for each horizon, h (Papadimitriou and Tsitsiklis 1987). Therefore, the total number of reachable beliefs in the GUSSP is $\mathcal{O}(|\Omega|^h)$, and thus the resulting belief MDP may have $\mathcal{O}(|\Omega|^h)$ distinct reachable states. □

In the worst case, the observation function may be unconstrained and all the beliefs may be unique. Since there is no discounting in a GUSSP and the horizon is unknown a priori, GUSSPs may have *infinitely* many beliefs and their complexity class may be undecidable in the worst case (Madani, Hanks, and Condon 1999). Hence, solving GUSSPs with non-myopic observations optimally is computationally intractable.

We now prove that a myopic observation function results in a finite number of reachable beliefs in a GUSSP.

Proposition 2. *A GUSSP with myopic observation function has a finite number of reachable beliefs.*

Proof. By definition, a myopic observation function produces either belief-collapsing observations or no information at all. For each case, we first calculate the updated belief for the goal configurations using Equation 3. Therefore, $\forall x' \in X$ with $x' = \langle s', g' \rangle$:

$$b'(g') = \frac{O(a, x', \omega)\, b(g)}{\sum_{x'} O(a, x', \omega) b(g)}.$$

Case 1: Belief-collapsing observation. Trivially, when $O(a, x', \omega) = 0$, the updated belief is $b'(g') = 0$. When $O(a, x', \omega) = 1$, the updated belief is $b'(g') = 1$.

Case 2: No information. When the observation provides no information, $\forall a \in A, O(a, x', \omega) = 1/|\Omega|$. Then,

$$b'(g') = \frac{b(g)/|\Omega|}{\sum_{x'} b(g)/|\Omega|} = b(g).$$

Thus, $\forall g \in G$, a myopic observation function produces collapsed belief or retains the same belief, resulting in a finite number of reachable beliefs for a goal configuration. Since $|S|$ is finite, the belief update following Equation 4 would result in finite number of reachable beliefs for a GUSSP. □

Hence, a myopic observation function weakly monotonically collapses beliefs, allowing us to simplify the problem further. We now show that a GUSSP reduces to an SSP, along the same lines as the mapping from a POMDP to belief-MDP (Kaelbling, Littman, and Cassandra 1998).

Proposition 3. *A GUSSP reduces to an SSP.*

Proof. We map the GUSSP to a belief MDP $\langle B, \mathcal{A}, \tau, \rho \rangle$ with a horizon h (Kaelbling, Littman, and Cassandra 1998), as in Proposition 1. By Proposition 2, a GUSSP with myopic observation function has a finite number of reachable beliefs and therefore, finite states in the belief-MDP. By construction, this belief-MDP is an SSP with the start state $\bar{s}_0 = b_0$ and the goal states, $\bar{S}_G$, are the set of states with $\bar{b}(x) = 1$ such that $\bar{b}(g) = 1$ and $s \in g$. Since there exists a proper policy in a GUSSP, the policy in this SSP is proper by construction. Thus, a GUSSP with myopic observation function reduces to an SSP. □

The reduction to an SSP facilitates solving GUSSPs using the existing rich suite of SSP algorithms. For ease of reference and clarity, we refer to the above-mentioned SSP as compiled-SSP in the rest of this paper.

The *order-k* of π^* for a GUSSP (compiled-SSP) can be calculated using a directed graph constructed using π^*. We now show that computing *order-k* is polynomial.

Proposition 4. *The worst case complexity for computing* order-k *for* π^* *is* $\mathcal{O}(|P_G|(|V|+|E|))$*, where* V *and* E *denote the vertices and edges of the corresponding directed graph.*

Proof Sketch. To calculate *order-*k for π^*, we construct a directed graph, Z, using π^* such that $V = \mathcal{I} \cup \{s_0\}$ and the trajectories between them are the edges, E. We begin with setting each potential goal to be a true goal. We introduce additional (artificial) edges from the true goal to the informative states. Then, we compute the strongly connected components, using depth first search that takes $\mathcal{O}(|V|+|E|)$, and condense it to form a directed acyclic graph $Z' = (V', E')$. We start from the true goal in Z' and traverse backwards. The k value of the true goal is initialized to 1 and propagated to its (unvisited) neighbors. At each vertex, k is increased to be the sum of informative states in the condensed vertex and the incoming value from the neighbor. This continues until all vertices in Z' have been visited and the start state is updated with the maximum k. This process may be repeated with every potential goal as the true goal and the overall maximum k is the order of the policy. Thus, the worst case complexity is $\mathcal{O}(|P_G|(|V| + |E|))$. □

Relation to Goal-POMDPs The Goal-POMDP (Bonet and Geffner 2009) models a class of goal-based and shortest-path POMDPs with positive action costs and no discounting. The set of target (or goal) states, $\bar{P}$, have unique belief-collapsing observations. Hence, a Goal-POMDP is a GUSSP when the partial observability is restricted to goals, the observations set is $2^{\bar{P}} \setminus \{\emptyset\}$, and observation function is myopic.

Proposition 5. *GUSSP* $\subset$ *Goal-POMDP.*

The observations in a Goal-POMDP are not constrained and may result in infinitely many reachable beliefs (Proposition 1). This makes it computationally challenging to compute optimal policies (Papadimitriou and Tsitsiklis 1987), unlike GUSSPs which are more tractable can be solved optimally (Proposition 3).

GUSSP with Deterministic Transitions A GUSSP with deterministic transitions presents an opportunity for further reduction in complexity. We show that the optimal policy in this case is a minimum spanning tree of its corresponding directed graph.

Proposition 6. *The optimal policy for a GUSSP with myopic observations and deterministic transitions is the minimum arborescence of a weighted and directed graph* Z*.*

Proof. Consider a GUSSP with deterministic transitions and a dummy start state, r, that transitions to the actual start state with probability 1 and zero cost. This can be represented as a directed and weighted graph, $Z = (V, E, w)$, such that $V = \{r\} \cup \{x \in X | x = \langle s, g \rangle \wedge s \in P_G\}$; that is, the start state and the potential goals are the vertices. Each edge $e \in E$ denotes a trajectory in the GUSSP between vertices. The proper policy in a GUSSP ensures that there is at least one edge between each pair of vertices. The weight of an edge connecting $x, y \in V$ is $w(e) = d(x, y)(1 - b(y))$, with $d(x, y)$ denoting the cost of the trajectory and $b(y)$ is the belief over y being a goal. The minimum arborescence (directed minimum spanning tree) of this graph, A, contains trajectories such that the total weight is minimized, $\min_{A \in \mathcal{A}} w(A)$ with $w(A) = \sum_{e \in A} w(e)$. By construction, this gives the optimal order of visiting the potential goals and hence the optimal policy for the GUSSP with $V^*(s_0) = w(A)$. □

Solving Compiled-SSPs

We propose (i) an admissible heuristic for SSP solvers that accounts for the goal uncertainty and (ii) a determinization-based approach for solving the compiled-SSP.

Admissible Heuristic

In heuristic search-based SSP solvers, the heuristic function helps avoid visiting states that are provably irrelevant. An efficient heuristic for solving the compiled-SSP guides the search by accounting for the goal uncertainty. We propose a heuristic for the compiled-SSP that accounts for goal uncertainty and is calculated as follows:

$$h_{pg}(x) \triangleq \min_{g \in G} \Big((1 - b(g)) \min_{i \in g} d(x, i) \Big)$$

where $d(x, i)$ denotes the cost of the shortest trajectory to the potential goal i from state x and $b(g)$ is the agent's belief of g being a true goal. Multiplying by the probability of a state not being a goal $(1 - b(g))$ breaks ties in favor of configurations with a higher probability of being a goal, with a lower heuristic value. The following proposition shows that the proposed heuristic is admissible.

Proposition 7. h_{pg} *is an admissible heuristic.*

Proof. To show that h_{pg} is admissible, we first show that $\min_{i \in g} d(x, i)$ is an admissible estimate of the expected cost of reaching a goal configuration g from state x. Let $d^*(x, g)$ be the expected cost of reaching g from x. Since $d(x, g)$ is the cost of the shortest trajectory to g from x, $d(x, g) \leq d^*(x, g)$. If all paths exist from x to all potential goal states $i \in g$, then by definition, the shortest trajectory to a goal configuration is the minimum distance to a potential goal in g. That is, $d(x, g) = \min_{i \in g} d(x, i)$ and therefore $\min_{i \in g} d(x, i) \leq d^*(x, g)$. Multiplying this value by the belief and using the minimum value over all possible goal configurations guarantees that h_{pg} is an admissible estimate of the expected cost reaching a true goal configuration. □

Determinization

Determinization is a popular approach for solving large SSPs as it simplifies the problem by replacing the probabilistic outcomes of an action with a single deterministic outcome (Yoon, Fern, and Givan 2007; Saisubramanian, Zilberstein, and Shenoy 2018). We extend determinization to a GUSSP by ignoring the uncertainty about the goals. The agent plans to reach one potential goal (determinized goal) at a time, simplifying the problem to a smaller SSP. During execution, if the determinized goal is not a true goal, the agent replans for another unvisited potential goal. This approximation scheme offers considerable speedup over solving the compiled-SSP.

	LAO* (Optimal solver)		Flares(1)-h_{min}		**Flares(1)-$\mathbf{h_{pg}}$**		**Det-MLG**		**Det-CG**	
Problem Instance	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time
rover (20,6)	28.25	14.99	35.35 ± 2.67	1.08	30.34 ± 2.37	0.17	36.71 ± 2.62	0.07	45.51 ± 3.22	0.06
rover (20,7)	42.16	30.19	43.49 ± 1.62	1.17	45.07 ± 1.77	0.83	49.69 ± 1.91	0.02	48.36 ± 1.43	0.03
rover (30,8)	36.96	190.92	38.21 ± 1.83	2.27	41.31 ± 1.97	0.16	38.54 ± 1.54	0.02	40.34 ± 1.82	0.03
rover (30,9)	34.72	832.56	38.21 ± 2.54	7.56	43.32 ± 2.54	1.73	50.27 ± 2.58	0.88	49.49 ± 1.97	0.45
search (20,4)	87.63	15.78	94.32 ± 0.58	1.45	93.32 ± 0.58	0.98	91.22 ± 0.67	1.05	90.42 ± 0.61	0.86
search (20,5)	74.61	14.42	83.83 ± 0.56	2.99	81.91 ± 0.56	1.93	78.32 ± 0.56	1.98	79.74 ± 6.37	0.98
search (20,5)	86.72	63.71	94.21 ± 0.79	6.21	91.18 ± 1.46	1.93	87.74 ± 0.65	0.66	89.98 ± 0.59	1.68
search (30,6)	90.89	267.35	94.21 ± 1.35	117.63	103.77 ± 3.42	21.07	101.67 ± 1.61	12.68	92.94 ± 0.68	19.50
ev (-,5)	2.34	8.16	3.29 ± 1.55	2.21	4.89 ± 1.36	0.92	5.15 ± 1.46	0.52	7.17 ± 1.43	0.62
ev (-,6)	3.46	10.79	4.89 ± 1.96	2.25	5.96 ± 1.96	1.14	7.15 ± 2.46	0.88	8.17 ± 1.43	0.79

Table 1: Average cost and planning time (seconds) results. Bold titles indicate our solution approaches.

We consider two determinization approaches: (i) most-likely goal determinization (DET-MLG) and (ii) closest-goal determinization (DET-CG). In the DET-MLG, the most-likely goal is determinized, based on its current belief. In DET-CG, the agent determinizes the closest goal based on the heuristic distance to the potential goal (with non-zero belief) from its current state. We resolve ties randomly.

Experiments

We begin with a comparison of different approximate solution techniques for solving the compiled-SSP on three domains in simulation. We then test the model on a real robot with three different initial belief settings.

Evaluation in Simulation

We experiment with three domains to evaluate the solution techniques in handling (i) location-based goal uncertainty (planetary rover domain, search and rescue domain) and (ii) temporal goal uncertainty (electric vehicle (EV) charging problem using real-world data). The expected cost of reaching the goal and run time are used as evaluation metrics. A uniform initial belief is considered for all the domains in these experiments. We solve the compiled-SSPs optimally using LAO* (Hansen and Zilberstein 1998; 2001), which is an optimal solver based on A* (Hart, Nilsson, and Raphael 1968) for solving MDPs with loops, and approximately using FLARES, a domain-independent state-of-the-art algorithm for solving large SSPs using horizon=1 (Pineda, Wray, and Zilberstein 2017), as well as the two determinization methods. The h_{min} heuristic, computed using a labeled version of LRTA* (Bonet and Geffner 2003), is used as a baseline for evaluating h_{pg}.

Planetary Rover This domain models the rover science exploration (Zilberstein et al. 2002; Ong et al. 2010) that explores an environment described by a known map to collect a mineral sample. The samples may be 'good' or 'bad' and $|P_G| = n$. The rover knows its own position (x, y) exactly, as well as those of the samples but does not know which samples are 'good'. The process terminates upon collecting a 'good' sample. The actions include moving in all four directions, which succeed with a probability of 0.8, and a *sample* action which is deterministic. The *sample* action costs $+2$ if the mineral is good and $+10$ otherwise; all other actions cost $+1$.

Search and Rescue In this domain, an autonomous robot explores an environment described by a known map to find victims (Pineda et al. 2015). We modify the problem such that there are m victims locations and n total victims. Each location may or may not have victims, which are known to the robot a priori. The state factors are the robot's current location and a counter to indicate the number of victims saved so far. The observations indicate the presence of victims in each state. The actions include moving in all four directions and a SAVE action that saves all the victims in a state. The move actions cost $+1$ and are stochastic, succeeding with 0.8 probability. The SAVE action is deterministic and costs $+2$. The objective is to minimize the expected cost of saving all victims.

Electric Vehicle Charging We experimented with the electric vehicle (EV) charging domain, operating in a vehicle-to-grid setting (Saisubramanian, Zilberstein, and Shenoy 2017), where the EV can charge and discharge energy from a smart grid. The objective is to devise a robust policy that is consistent with the owner's preferences, while minimizing the operational cost of the vehicle. We modified the problem such that parking duration of the EV is uncertain with H denoting the horizon. The potential goals in this problem are the possible departure times. The EV can fully observe its current charge level and the time step. In our experiments, $|P_G| = n$ denotes that $P_G = \{H, H-1, .., H-n\}$. Each t is equivalent to 30 minutes in real time. If the EV's exit charge level does not meet the owner's desired exit charge level, a penalty is incurred.

The battery capacity and the charge speeds for the EV are based on Nissan Leaf configuration and the action costs and peak hours are based on real data (Eversource 2017). The charge levels and entry time data are based on charging schedules of electric cars over a four month duration in 2017 from a university campus. The data is clustered based on the entry and exit charges, and we selected 25 representative problem instances across clusters for our experiments.

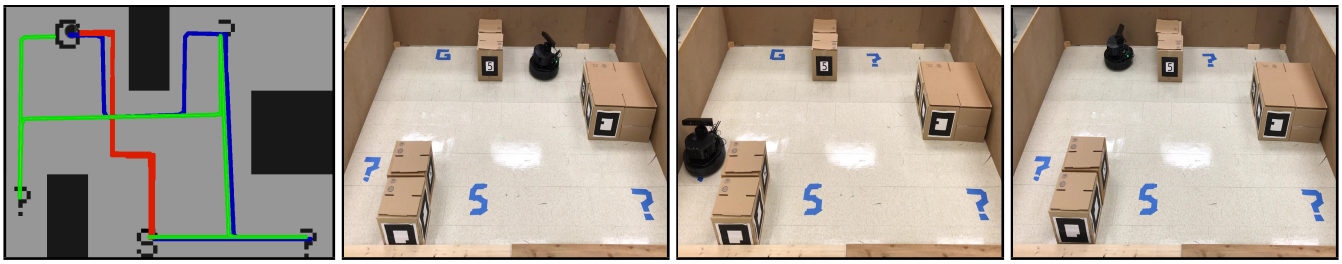


Figure 3: Demonstration of the path taken by the robot with three different initial beliefs for the map in Figure 1. The start state and the true goal state are denoted by S and G, respectively. The other potential goals are denoted by the question mark symbol. Green, blue, and red show the path taken by the robot with 0.1, 0.25, and 0.9 as the initial belief for the true goal state and equal probability for other potential goal states.

Discussion Table 1 shows the results of the five techniques on various problem instances, in terms of cost and runtime respectively. The results are averaged over 100 trials and standard errors are reported for the expected cost. The results for the EV domain are averaged over 25 problem instances. The grid size and the number of potential goals are shown for each problem instance. We experiment with no landmark states to demonstrate the performance in the worst case setting. The significance threshold was set at 10%. In terms of expected costs, the performance of the approximate techniques are comparable. The runtimes for solving the problems optimally, however, scales rapidly as the number of potential goals increases. The advantage of using FLARES with h_{pg} and the determinization techniques are more evident in the runtime savings. FLARES using our heuristic h_{pg} is significantly faster than using the baseline h_{min} heuristic. The determinizations are faster than solving the problem using FLARES with either heuristic.

Evaluation on a mobile robot

The robot experiment aims to visually explain how the belief distribution alters the robot's trajectory. Figure 3 shows the results in a ROS simulation and on a real robot for a simple search and rescue problem with one agent and four potential victim locations for the map shown in Figure 1. We test with three different initial beliefs: uniform, optimistic, and pessimistic. The corresponding belief of the true goal, G, in each belief setting is: 0.25, 0.9, and 0.1, with the other potential goals having equal probability. The *order-k* of the optimal policy with respect to the true goal in each belief setting is 4, due to stochastic transitions. The *order-k* for the optimal policies of the GUSSP with deterministic transitions for this problem are: 3, 1, and 4, corresponding to the three initial beliefs.

Conclusion and Future Work

The goal uncertain SSP (GUSSP) provides a natural model for real-world problems where it is non-trivial to identify the exact goals ahead of plan execution. While a general GUSSP could be intractable, we identify several tractable classes of GUSSPs and propose effective algorithms for solving them. Specifically, we show that a GUSSP with a myopic observation function can be reduced to an SSP, allowing us to efficiently solve it using existing SSP solvers. We also propose an admissible heuristic that accounts for goal uncertainty in its estimation and a fast solver based on extending the notion of determinization to handle goal uncertainty. The simulation results show that solving the compiled-SSPs using FLARES with the proposed heuristic is faster than the baseline. The determinization techniques are significantly faster than solving the compiled-SSP optimally. The results show that GUSSPs can be solved efficiently using scalable algorithms that do not rely on POMDP solvers. In the future, we aim to explore other conditions under which GUSSPs have a bounded set of beliefs that supports the development of efficient solvers.

Acknowledgments

This work was supported in part by the National Science Foundation grants IIS-1524797 and IIS-1724101.

References

Amato, C., and Zilberstein, S. 2009. Achieving goals in decentralized POMDPs. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*.

Bertsekas, D. P., and Tsitsiklis, J. N. 1991. An analysis of stochastic shortest path problems. *Mathematics of Operations Research,* 16:580–595.

Biswas, J., and Veloso, M. 2013. Multi-sensor mobile robot localization for diverse environments. In *Robot Soccer World Cup*, 468–479. Springer.

Bonet, B., and Geffner, H. 2003. Labeled RTDP: Improving the convergence of real-time dynamic programming. In *Proceedings of the 13th International Conference on Automated Planning and Scheduling*.

Bonet, B., and Geffner, H. 2009. Solving POMDPs: RTDP-bel vs. point-based algorithms. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*.

Bourgault, F.; Furukawa, T.; and Durrant-Whyte, H. F. 2003. Coordinated decentralized search for a lost target in a bayesian world. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*.

Delgado, K. V.; Sanner, S.; De Barros, L. N.; and Cozman, F. G. 2011. Efficient solutions to factored MDPs with imprecise transition probabilities. *Artificial Intelligence* 175:1498–1527.

Eversource. 2017. Time of use rates. https://www.eversource.com/clp/vpp/vpp.aspx.

Hansen, E. A., and Zilberstein, S. 1998. Heuristic search in cyclic AND/OR graphs. In *Proceedings of the 15th National Conference on Artificial Intelligence*.

Hansen, E. A., and Zilberstein, S. 2001. LAO*: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence* 129:35–62.

Hansen, E. A. 2007. Indefinite-horizon POMDPs with action-based termination. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* 4:100–107.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101:99–134.

Kitano, H.; Tadokoro, S.; Noda, I.; Matsubara, H.; Takahashi, T.; Shinjou, A.; and Shimada, S. 1999. RoboCup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *IEEE Conference on Systems, Man, and Cybernetics*.

Littman, M. L. 1997. Probabilistic propositional planning: Representations and complexity. In *Proceedings of the 14th Conference on Artificial Intelligence*.

Madani, O.; Hanks, S.; and Condon, A. 1999. On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems. In *Proceedings of the 16th AAAI Conference on Artificial Intelligence*.

Nie, X.; Wong, L. L.; and Kaelbling, L. P. 2016. Searching for physical objects in partially known environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

Ong, S.; Png, S. W.; Hsu, D.; and Lee, W. S. 2010. Planning under uncertainty for robotic tasks with mixed observability. *International Journal of Robotics Research* 29:1053–1068.

Papadimitriou, C. H., and Tsitsiklis, J. N. 1987. The complexity of markov decision processes. *Mathematics of Operations Research* 12:441–450.

Patek, S. D. 2001. On partially observed stochastic shortest path problems. In *Proceedings of the 40th IEEE Conference on Decision and Control*.

Pineda, L.; Takahashi, T.; Jung, H.-T.; Zilberstein, S.; and Grupen, R. 2015. Continual planning for search and rescue robots. In *Proceedings of the 15th IEEE Conference on Humanoid Robots*.

Pineda, L.; Wray, K.; and Zilberstein, S. 2017. Fast SSP solvers using short-sighted labeling. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

Saisubramanian, S.; Zilberstein, S.; and Shenoy, P. 2017. Optimizing electric vehicle charging through determinization. In *Scheduling and Planning Applications Workshop (SPARK), ICAPS*.

Saisubramanian, S.; Zilberstein, S.; and Shenoy, P. 2018. Planning using a portfolio of reduced models. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*.

Stone, L. D.; Royset, J. O.; and Washburn, A. R. 2016. Search for a stationary target. In *Optimal Search for Moving Targets*. Springer. 9–48.

Trevizan, F., and Veloso, M. 2013. Finding objects through stochastic shortest path problems. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*.

Yoon, S.; Fern, A.; and Givan, R. 2007. FF-Replan: A baseline for probabilistic planning. In *Proceedings of the 17th International Conference on Automated Planning and Scheduling*.

Zilberstein, S.; Washington, R.; Bernstein, D. S.; and Mouaddib, A.-I. 2002. Decision-theoretic control of planetary rovers. In *Revised Papers from the International Seminar on Advances in Plan-Based Control of Robotic Agents*, 270–289. Springer-Verlag.