Adaptive Informative Path Planning with Multimodal Sensing

Shushman Choudhury,[*1] **Nate Gruver,**[*1] **Mykel J. Kochenderfer**[2]
[1]Computer Science, Stanford University
[2]Aeronautics & Astronautics, Stanford University
{shushman, ngruver, mykel}@stanford.edu

Abstract

Adaptive Informative Path Planning (AIPP) problems model an agent tasked with obtaining information subject to resource constraints in unknown, partially observable environments. Existing work on AIPP has focused on representing observations about the world as a result of agent movement. We formulate the more general setting where the agent may choose between different sensors at the cost of some energy, in addition to traversing the environment to gather information. We call this problem AIPPMS (MS for Multimodal Sensing). AIPPMS requires reasoning jointly about the effects of sensing and movement in terms of both energy expended and information gained. We frame AIPPMS as a Partially Observable Markov Decision Process (POMDP) and solve it with online planning. Our approach is based on the Partially Observable Monte Carlo Planning framework with modifications to ensure constraint feasibility and a heuristic rollout policy tailored for AIPPMS. We evaluate our method on two domains: a simulated search-and-rescue scenario and a challenging extension to the classic RockSample problem. We find that our approach outperforms a classic AIPP algorithm that is modified for AIPPMS, as well as online planning using a random rollout policy.

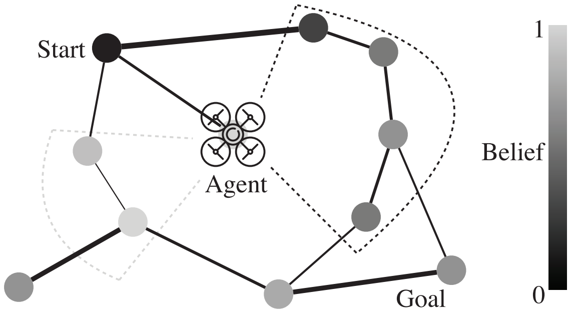

Figure 1: The AIPPMS problem requires an agent to plan paths over a graph of locations in an environment, while maximizing the information gathered by visiting locations subject to an energy constraint (the edge thickness represents the energy cost of traversal). The states of locations (binary in this example) are observed through noisy sensor readings. The agent is equipped with two kinds of sensors with different fidelity and range parameters. The shade of the node indicates the current belief of the agent about its state.

1 Introduction

For various robotic applications such as search-and-rescue, terrain exploration, and environmental monitoring, an autonomous agent must gather information in uncertain environments with partial observability. The agent is equipped with multiple sensing modalities with which it receives noisy observations of the world. It also has limited energy, which is expended both by using the sensors and by movement. We call this the problem of Adaptive Informative Path Planning with Multimodal Sensing (AIPPMS). In such settings, there is a trade-off between exploring the environment and exploiting the current belief about the environment to visit informative locations. Our objective is to obtain an adaptive strategy that guides the agent from the start to the goal location while balancing this exploration-exploitation tradeoff and respecting the energy budget constraint.

AIPPMS can model a variety of real-world problems. For instance, in a search-and-rescue mission, one or more robots must traverse unstructured terrain, detect the presence of survivors with noisy sensors and then decide which locations to visit in order to save as many survivors as possible within their onboard battery life (Singh, Krause, and Kaiser 2009). Other domains include monitoring sensitive ecosystems (Das et al. 2015) and deploying remote devices to cover an unknown landscape (Binney and Sukhatme 2012).

The informative path planning (IPP) problem is NP hard in the non-adaptive setting (Meliou et al. 2007), where the agent plans and executes a path without accounting for noisy observations. The adaptive setting adds another dimension of difficulty by requiring not just a plan but a policy that reacts to the observations the agent receives. We generalize the problem even further by explicitly introducing multiple sensing modalities and considering the effect of both sensing and moving on the energy budget.

Prior work has focused on non-adaptive IPP with various approximation algorithms (Singh et al. 2007). The adaptive setting has been studied with utility function assumptions such as submodularity and locality (Singh, Krause, and Kaiser 2009), hypothesis identification in metric spaces (Lim,

[*] Equal contribution.

Hsu, and Lee 2016), and a data distribution over possible environments (Choudhury et al. 2017). To the best of our knowledge, our multimodal sensing formulation has not been considered in previous work on adaptive IPP. Unlike our problem, in previous formulations the only decision is the location to visit next, observations are received on visiting locations, and only movement depletes energy.

Our key idea is to *reason jointly about multimodal sensing and movement* by formulating the problem as a Partially Observable Markov Decision Process (POMDP) and using an online solver to accommodate the state and observation spaces, which are exponentially large (in the number of locations). We use constrained search in the online solver to ensure that the policy always satisfies the goal and energy constraints. For a tailored rollout policy to guide the online simulations, we use an adaptive extension to a near-optimal algorithm for the fully observable variant of our problem (Zhang and Vorobeychik 2016). Through this online POMDP framework, we obtain an adaptive IPP algorithm that judiciously balances exploration with multimodal sensing and exploitation with movement. Our approach consistently outperforms a nonmyopic adaptive IPP algorithm (Singh, Krause, and Kaiser 2009) and our ablation study shows that the tailored rollout policy is key to this improvement.

2 Background

This section provides relevant background on the Informative Path Planning problem, Partially Observable Markov Decision Processes (POMDPs), and online POMDP solvers.

2.1 Informative Path Planning

In informative path planning (IPP) we must choose the best subset of locations to visit (thereby gathering information) subject to constraints on the path, such as energy expended. The IPP problem is relevant to various mobile robotics applications such as mapping with terrestrial and aerial vehicles (Heng et al. 2015; Charrow et al. 2015) and adaptive sampling for underwater environments (Binney, Krause, and Sukhatme 2010).

The IPP problem is NP-hard, as is path planning in general (Canny 1987). IPP can be framed as an orienteering problem (Golden, Levy, and Vohra 1987), which is a generalization of the known NP-hard Traveling Salesman Problem with an additional constraint on how far the agent can travel. A number of efficient heuristics and approximation techniques have been explored. Gaussian Processes were used to model the environment and the mutual information between locations (Singh et al. 2009), and minimum-cost tours have been computed on efficiently chosen subsets of nodes (Hollinger and Sukhatme 2013).

The adaptive IPP (AIPP) problem is even more challenging because we seek a reactive policy that chooses the next location to visit based on the observations so far. Initial work on AIPP used information theoretic arguments with myopic heuristics (Stachniss, Grisetti, and Burgard 2005). A non-myopic approach for AIPP (Singh, Krause, and Kaiser 2009) provides performance guarantees when the utility function is monotone submodular and locations far apart are weakly independent. When the environment task is to narrow down a hypothesis set of possible worlds, an efficient method has been developed with Group Steiner trees (Lim, Hsu, and Lee 2016). Finally, when a prior over informative locations is available, imitation of an oracle achieves good performance while also providing useful theoretical guarantees (Choudhury et al. 2017; 2018).

2.2 POMDPs

POMDPs provide a principled mathematical framework for sequential decision-making under uncertainty (Kochenderfer 2015). A POMDP is typically represented by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, T, Z, R, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, and $\mathcal{O}$ is the observation space. The transition function T maps states and actions to a distribution over next states, i.e., $T(s, a, s') = P(s' \mid s, a)$. When an agent executes an action in a state, it receives a noisy observation of the state modeled by the observation function Z, i.e. $Z(s, a, o) = P(o \mid s, a)$.

The reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ specifies the expected one-step reward received by the agent $R(s, a)$ upon taking action a in state s. The discount factor $\gamma \in [0, 1)$ is provided for infinite-horizon problems to ensure that the total expected reward over a trajectory $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$ is bounded when the rewards are bounded. In a POMDP, states are not directly observable. We typically work with the belief space $\mathcal{B}$, which is a space of probability distributions over $\mathcal{S}$. The history of actions and observations can be entirely captured in the current belief state. A POMDP *policy* $\pi : \mathcal{B} \to \mathcal{A}$ maps the belief state to an action to take.

The objective of solving a POMDP is to obtain an optimal policy, i.e. one that maximizes the expected cumulative reward. Exact solution methods for both finite horizon (Smallwood and Sondik 1973) and discounted infinite horizon (Sondik 1978) cases are well-established but intractable in general (Papadimitriou and Tsitsiklis 1987). Consequently, recent efforts have focused on developing approximate solutions (Hauskrecht 2000).

2.3 Online POMDP Planning

Online approaches to POMDPs choose actions at runtime by reasoning over a limited future horizon of belief states reachable from the current belief state. A survey of online methods (Ross et al. 2008) outlines popular approaches, such as branch-and-bound and forward search.

Two recent state-of-the-art methods for online planning are POMCP (Silver and Veness 2010) and DESPOT (Ye et al. 2017). The first of these is based on Monte Carlo Tree Search with Upper Confidence Bounds for exploratory actions, while the second uses a sparse approximation of the belief tree rooted at the current belief state. We use the POMCP framework for AIPPMS because it is simpler. Our modifications can also be used with variations of POMCP, such as POMCPOW (Sunberg and Kochenderfer 2018) for continuous action spaces. A very recent online approach attempts to address the exploration-exploitation trade-off in informative planning by Pareto-optimal Monte Carlo Tree Search (Chen and Liu 2019) but does not allow for multimodal sensing.

3 Problem Definition

The problem that we consider is Adaptive Informative Path Planning with Multimodal Sensing (AIPPMS). We will use notation consistent with previous established work on Adaptive IPP (Singh, Krause, and Kaiser 2009) and its relation to POMDPs (Choudhury et al. 2017).

We have a location graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set of nodes $\mathcal{V}$ corresponds to all discrete locations the agent can visit and sense. An edge $e = (u, v) \in \mathcal{E}$ has weight $C(u, v)$ equal to the cost of traveling between locations u and v. The robot starts at v_s and needs to reach a given goal node v_g. Each node v has some true **state** represented by a random variable $\mathcal{X}_v$. The set of all possible states of a node is $\mathcal{X}$ and the random vector $\mathcal{X}_\mathcal{V} = (\mathcal{X}_1, \ldots, \mathcal{X}_n)$ defines the unobserved true state of the environment. For instance, in the disaster rescue scenario, $\mathcal{X}_\mathcal{V}$ represents the density of survivors at the various pickup locations. The dependencies between location states is modeled by joint distribution $P\left(\mathcal{X}_\mathcal{V}\right)$.

The agent has a suite of sensors $\mathcal{S}$, with each sensor σ having some usage cost $C(\sigma)$. Let $y \in \mathcal{Y}$ be an **observation** received by the robot. Let $H : \mathcal{V} \times \mathcal{X}^{|\mathcal{V}|} \times \mathcal{S} \to \mathcal{Y}$ be the observation function. When the robot is at node v in a environment of state $\mathcal{X}_\mathcal{V}$ and uses sensor σ, the measurement y received by the robot is $y = H\left(v, \mathcal{X}_\mathcal{V}, \sigma\right)$. The sensor observation model $P\left(\mathcal{Y} = y \mid \mathcal{V} = v, \mathcal{S} = \sigma, \mathcal{X}_\mathcal{V}\right)$ is stochastic but the form is known. The sensors typically have varying combinations of performance (fidelity; range) and energy consumption (details in Section 5).

A valid **action** at the current node v_t is to either go to a different node v_{t+1} or to use some sensor σ. The movement between locations (and corresponding energy expended) is fully deterministic. The **cost** of an action a_t taken at current location v_t is therefore either some travel cost to a different node or the energy cost of sensing at that node, i.e.,

$$C(v_t, a_t) = \begin{cases} C(v_t, v_{t+1}) & \text{if } a_t = v_{t+1},\ v_{t+1} \neq v_t \\ C(\sigma) & \text{if } a_t = \sigma,\ v_{t+1} = v_t \end{cases}. \quad (1)$$

Let $F : 2^\mathcal{V} \times \mathcal{X}^{|\mathcal{V}|} \to \mathbb{R}_{\geqslant 0}$ be a **utility** function mapping a subset of nodes which have been visited and a world state to some utility which represents the information in the environment. For a collection of visited nodes ξ and a world map encoded in $\mathcal{X}_\mathcal{V}$, $F\left(\xi, \mathcal{X}_\mathcal{V}\right)$ assigns a utility. Note that F is a set function. After a node has been visited, visiting it again adds no further utility, but sensing at a visited node may be used as an information gathering action to refine the belief of the true state at other nodes, based on the observation function updates. Given a node $v \in \mathcal{V}$, a set of nodes $V \subseteq \mathcal{V}$ and world $\mathcal{X}_\mathcal{V}$, the discrete derivative of the utility function F is $\Delta_F\left(v \mid V, \mathcal{X}_\mathcal{V}\right) = F\left(V \cup \{v\}, \mathcal{X}_\mathcal{V}\right) - F\left(V, \mathcal{X}_\mathcal{V}\right)$.

The **objective** of the general Adaptive IPP problem is to maximize the expected utility of visited nodes, starting at v_s and ending at v_g, subject to some cost budget B. Note that the *utility* function F which is relevant to the objective, is entirely distinct from the *cost* function C which is relevant to the constraint. The location selection is done by a policy π which, at each timestep t, maps the observations received $\{y_i\}_{i=1}^{t-1}$ and locations visited $\{v_i\}_{i=1}^{t-1}$ to the node to visit v_t. In our AIPPMS problem, the policy is subject to the same energy constraint, but it can choose to perform a sensing action at timestep t instead of going to another location, while expending some energy cost for sensing. In resource-constrained problems where only a few locations can be visited, sensing can be an important means of determining which locations are likely to have greater utility. *Note that sensing itself has no intrinsic utility in terms of F.*

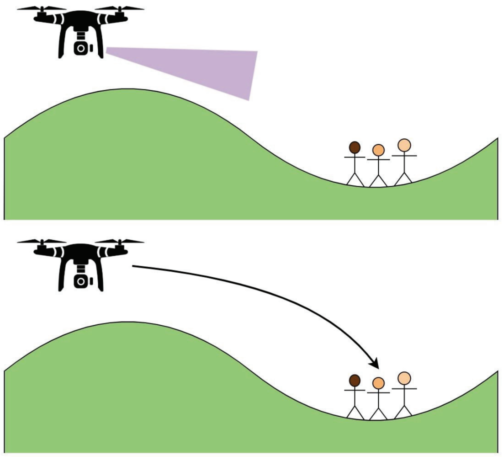

Figure 2: The AIPPMS formulation allows the agent to reason about modulating or completely turning off sensing to conserve energy when it has sufficient information. The upper panel shows how the agent (in a traditional AIPP setting) continues to sense despite sufficient information. The lower panel shows the agent directly moving to the next location without potentially expensive sensing.

4 Approach

Our problem inherits all the complexities of Adaptive IPP but adds the additional complexity of deciding between sensors with varying energy costs and observation models. Because of its inherently probabilistic nature, this problem can be framed quite naturally as a POMDP. Despite this applicability, most relevant approaches to AIPP (Singh, Krause, and Kaiser 2009; Lim, Hsu, and Lee 2016; Choudhury et al. 2017) *do not explicitly solve AIPP as a POMDP*. While each approach states their own reasons for this, an implicit reason is the generality of POMDPs, which does not allow the exploiting of AIPP-specific assumptions. Previous approaches, for example, separately address the utility of an action, which affects the objective, and the feasibility of an action, which affects the energy constraint.

In the more general AIPPMS, however, decomposability is greatly reduced. The agent must choose between sensing and moving at each timestep, and multiple sensing modalities have different trade-offs between energy and expected information gain. A particularly evocative example is shown in Figure 2, where turning off sensing limits energy usage. *We choose to explicitly formulate and solve AIPPMS as a POMDP in order to jointly reason about the effect of movement and multimodal sensing actions.* We describe this formulation in Section 4.1 and then describe in Section 4.2 how we use a online POMDP solver tailored to AIPPMS.

4.1 POMDP Formulation

We now formalize AIPPMS as a POMDP using the notation of Section 3. The AIPPMS is a discrete-time, finite-horizon, constrained POMDP. The **state** at time t is defined as $s_t = (v_t, \xi_t, \Delta e_t, \mathcal{X}_{\mathcal{V}})$ where v_t is the agent's current node location, ξ_t is the set of nodes that have been visited already, $\Delta e_t = B - \sum_{i=1}^{t-1} C(s_i, a_i)$ is the remainder of the cost budget. These three components are all fully observable. The world state $\mathcal{X}_{\mathcal{V}}$ is defined by the specific problem but is not observable. The **belief** is over the most likely state of the world. In the most general sense, the belief space $\mathcal{B} \subset \mathcal{X}^{|\mathcal{V}|}$, but restrictive assumptions on the joint distribution can be made for tractability; *the modeling of the joint distribution between locations is agnostic to the POMDP solver*.

The full **action** space is the union of all locations and sensors, $\mathcal{A} = \mathcal{V} \cup \mathcal{S}$, but the set of valid actions for a state depends on the neighbours of the current node in the location graph, $\mathcal{A}(s_t) = \mathrm{Nbrs}(v_t) \cup \mathcal{S}$. The **observation** space is obtained from the AIPPMS specification, $\mathcal{O} = \mathcal{Y}$. An example observation is just the states of some subset of locations, $o_t = \{\hat{\mathcal{V}}, \mathcal{X}_{\hat{\mathcal{V}}}\}$ where $\hat{\mathcal{V}} \subseteq \mathcal{V}$. For sensing actions, the observations are noisy and depend on the domain-specific sensor observation model. For movement actions, the observation is deterministic and is just the true state of the visited location.

The **transition** function T is fully deterministic. If a_t is a movement action to a neighboring node, the new state is $s_{t+1} = (v_{t+1}, \xi_{t+1}, \Delta e_{t+1}, \mathcal{X}_{\mathcal{V}})$ where $v_{t+1} = a_t$, $\xi_{t+1} = \xi_t \cup v_{t+1}$ and $\Delta e_{t+1} = \Delta e_t - C(s_t, a_{t+1})$. If a_t is a sensing action, the only difference between s_t and s_{t+1} is for $\Delta e_{t+1} = \Delta e_t - C(s_t, a_{t+1})$. The belief state over the world is updated with the observation from the sensor as follows:

$$\begin{aligned} & b_{t+1} = \tau(b_t, o_t, \sigma) \\ \Longrightarrow\ & b_{t+1}(\mathcal{X}'_{\mathcal{V}}) \propto P(o_t \mid \mathcal{X}'_{\mathcal{V}}, \sigma, b_t)\, P(\mathcal{X}'_{\mathcal{V}} \mid \sigma, b_t) \\ & \propto Z(\mathcal{X}'_{\mathcal{V}}, \sigma, o_t) \sum_{\mathcal{X}_{\mathcal{V}}} T(\sigma, \mathcal{X}_{\mathcal{V}}, \mathcal{X}'_{\mathcal{V}})\, P(\mathcal{X}_{\mathcal{V}} \mid b_t) \\ \Longrightarrow\ & b_{t+1}(\mathcal{X}'_{\mathcal{V}}) \propto Z(\mathcal{X}'_{\mathcal{V}}, \sigma, o_t)\, b_t(\mathcal{X}'_{\mathcal{V}}) \end{aligned} \tag{2}$$

which is standard recursive Bayesian estimation of the state of the world. Since the world state is assumed to be fixed, this update can be done more efficiently than in the general case where the state changes. However, the efficiency of this update depends on how the joint distribution over the node states $P(\mathcal{X}_{\mathcal{V}})$ is maintained; in general, MAP inference with belief networks is NP-hard (Shimony 1994).

The **reward** function for the POMDP is defined in terms of the AIPPMS utility function F as follows:

$$R(s_t, a_t, s_{t+1}) = \begin{cases} 0 & \text{if } a_t = \sigma \\ \Delta_F(v_{t+1} \mid \xi_t, \mathcal{X}_{\mathcal{V}}) & \text{if } a_t = v_{t+1} \end{cases} \tag{3}$$

Therefore, an agent receives reward when it visits a node and observes its true underlying state. This reward is different from the action's cost, which is obtained from (1). The problem terminates when the agent has too little energy left to take another action, i.e. $\Delta e_t < \min_a C(s_t, a)$. If the agent is not at the goal v_g when this happens, it receives a reward of $-\infty$. If it is at the goal, the cumulative reward is the utility of all the visited nodes. The deterministic nature of the energy cost function allows us to define a state-dependent feasible set of actions that do not violate the energy constraint. We will discuss subsequently how we incorporate this feasible action set in our specific online planning framework.

4.2 Online Planning for AIPPMS

We have described in Section 2.2 the computational challenges involved in solving POMDPs, and the motivation for online planning in large domains with substructure in the reachability of states. For AIPPMS, the connectivity of the graph $\mathcal{G}$ makes only certain states reachable from other states (the possible values of $v_{t+1},\ \xi_{t+1}$, and Δe_{t+1} given their values at t are restricted by the graph). Furthermore, both state and observation spaces are exponential in the number of nodes, making an online planning approach preferable.

We use Partially Observable Monte Carlo Planning or POMCP (Silver and Veness 2010) as the underlying online solver. We tailor POMCP to solve AIPPMS problems with two specifications, that we now describe. First, we prune all constraint-violating actions during the lookahead search from the current state. Second, we develop a rollout policy that is quite suitable for a relevant class of utility functions.

Action Pruning The constraint for AIPPMS is to reach the goal vertex v_g within the cost budget B. Due to the deterministic behavior of the constraint cost, we can exactly specify a feasibility condition on any state of the POMDP. For the location graph $\mathcal{G}$, let the shortest path between any two vertices u and v on the graph be denoted as $C_{\mathcal{G}}(u, v)$. Then, a state $s_t = (v_t, \xi_t, \Delta e_t, \mathcal{X}_{\mathcal{V}})$ is feasible if $\Delta e_t > C_{\mathcal{G}}(u, v)$, i.e. the agent has sufficient energy in that state to go to the goal.

Denote the set of all feasible states as $\bar{\mathcal{S}}$. Then, for any feasible state $s \in \bar{\mathcal{S}}$, the set of feasible actions comprises those that can only lead to another feasible state, i.e.

$$\bar{\mathcal{A}}(s) = \{a \in \mathcal{A} \mid T(s, a, s') > 0 \Rightarrow s' \in \bar{\mathcal{S}}\} \tag{4}$$

and can be computed efficiently for any state by caching and looking up the all-pairs shortest paths matrix for $\mathcal{G}$ using Floyd-Warshall's algorithm (Floyd 1962). An illustration with a particular scenario is shown in Figure 3. The action pruning enables more exhaustive searches for the same planning time.

GCB Rollout Policy For POMCP, the rollout policy is used in the second stage of simulations to estimate the value of a leaf node in the search tree. In principle, the rollout policy could be an uninformed one that chooses actions at random, but in practice, the choice of rollout can greatly impact the performance of POMCP on large problems.

We use a rollout policy based on the Generalized Cost-Benefit or GCB Algorithm (Zhang and Vorobeychik 2016), which is designed for the fully observable IPP problem (when all utilities of visiting locations are known). The performance guarantees of GCB only hold for problems where the utility function is submodular, i.e. obeys the property of diminishing returns (Nemhauser, Wolsey, and Fisher 1978). This is not an

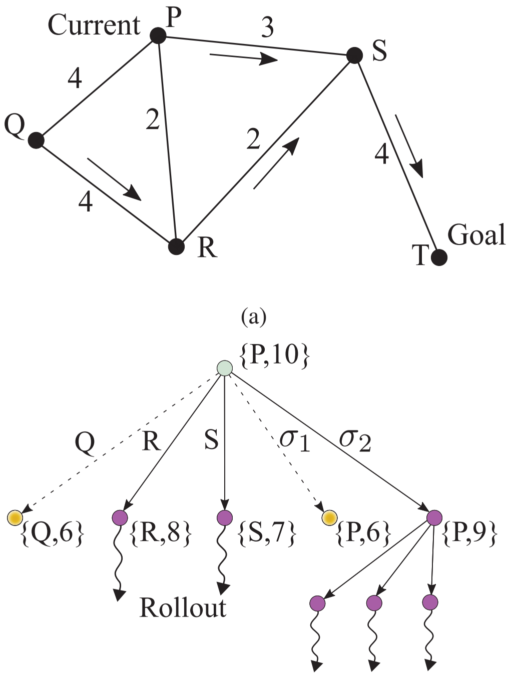

Figure 3: Constrained action selection in our modified POMCP. (a) An example scenario with arrows representing the direction of the shortest cost path. (b) Action pruning resulting from the energy budget – dotted line actions are never taken, while rollouts continue down bold paths.

issue for us because submodular functions are quite prevalent in real-world settings (Krause and Golovin 2014) and particularly in informative path planning (Meliou et al. 2007; Binney, Krause, and Sukhatme 2010). The key idea of GCB is to greedily choose the next action or option that maximizes the ratio of marginal utility or benefit to cost expended. For our partially observable stochastic optimization case, we use an adaptive greedy strategy that computes expected marginal utility (Golovin and Krause 2011).

The adaptive greedy rollout policy is outlined in the ACTION procedure in Algorithm 1. As before, we only consider the set of feasible actions for the state. For movement actions ($a \in \mathcal{V}$), the expected marginal utility is computed based on the belief state $b(\mathcal{X}_\mathcal{V})$ which is equivalently encoded in the history h. Sensing actions ($a \in \mathcal{S}$) have no utility with respect to the AIPPMS objective F, but we incentivize them in the rollout with an information-theoretic reward,

$$\begin{aligned}\mathrm{IG}(a|b) &= \sum_o P(o \mid b, a)\left[\max_s \tau(b,o,a)(s) - \max_s b(s)\right] \\ &\approx \sum_i^{\#\text{ samples}} \left[\max_s \tau(b,o_i,a)(s) - \max_s b(s)\right]\end{aligned} \tag{5}$$

Algorithm 1 POMCP with GCB Rollout for AIPP-MS

```
procedure SIMULATE($s$, $u$, $depth$)
    if $\gamma^{depth} < \epsilon$
        return 0
    if $h \notin T$
        for all $a \in \bar{\mathcal{A}}(s)$
            $T(ha) \leftarrow (N_{init}(ha), V_{init}(ha), \varnothing)$
        return ROLLOUT($s, h, depth$)    ▷ This uses ACTION internally
    $a \leftarrow \operatorname*{argmax}_{b \in \bar{\mathcal{A}}(s)} V(hb) + c\sqrt{\frac{\log N(h)}{N(hb)}}$
    $(s', o, r) \sim \mathcal{G}_{\text{sim}}(s, a)$
    $R \leftarrow r + \gamma \cdot \text{SIMULATE}\,(s', hao, depth + 1)$
    $B(h) \leftarrow B(h) \cup \{s\}$
    $N(h) \leftarrow N(h) + 1$
    $N(ha) \leftarrow N(ha) + 1$
    $V(ha) \leftarrow V(ha) + \frac{R - V(ha)}{N(ha)}$
    return $R$

procedure ACTION($\pi_{rollout}$,$s$,$h$)    ▷ GCB Rollout
    for all $a \in \bar{\mathcal{A}}(s)$
        if $a \in \mathcal{V}$    ▷ $a$ is for movement
            $U(a) \leftarrow \mathbb{E}_{b(\mathcal{X}_\mathcal{V})}\left[\Delta_F\left(a \mid \xi, \mathcal{X}_\mathcal{V}\right)\right]/C(s,a)$
        else    ▷ $a$ is for sensing
            $U(a) \leftarrow \mathrm{IG}\left(a \mid b(\mathcal{X}_\mathcal{V})\right)/C(a)$
    return $a \sim \mathrm{SoftMax}(U)$
```

Sensing actions are thus selected to maximize in expectation the mode of the belief state. This technique combines an expected information gain approach (Stachniss, Grisetti, and Burgard 2005) with the insight that distribution mode can be used as a lightweight approximation of negative information entropy (Dressel and Kochenderfer 2017), as collapsed distributions necessarily have more concentration of density. Finally, having computed the expected cost-benefit ratio for each of the feasible actions from the state, we sample actions from a softmax distribution over these values.

5 Experiments

We run all simulations in the Julia programming language for its fast numerical computations (Bezanson et al. 2017). We used the POMDPs.jl framework for specifying our POMDP formulation and for the base implementation of POMCP (Egorov et al. 2017).

5.1 Baselines

Since we are introducing AIPPMS in this work, there is no existing baseline we could compare against directly. We therefore extend the NAIVE or Nonmyopic Adaptive InformatiVE path planning algorithm (Singh, Krause, and Kaiser 2009), which is an elegant and theoretically motivated approach for AIPP, to our AIPPMS problem. The NAIVE algorithm consists broadly of iterative Bayesian updating (the same as for the modified POMCP) and replanning using a nonadaptive method called *pSPIEL-Orienteering* or PSPIEL_{OR}. We omit an elaboration of NAIVE here and refer readers to its paper. *NAIVE is the acronym of the algorithm and is not meant to indicate that the baseline is actually naive.*

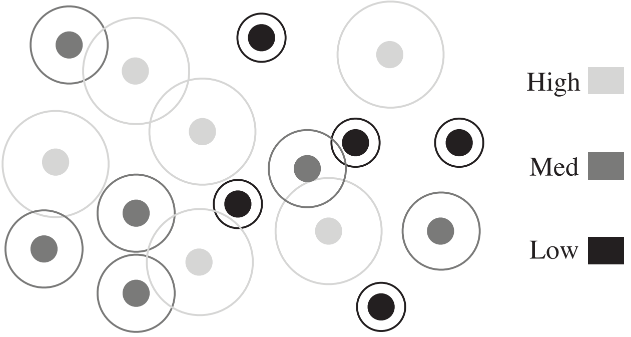

Figure 4: An illustration of the utility in the search-and-rescue problem scenario used for our experiments. Each node in the graph has a state corresponding to one of high, medium or low accessibility to survivors. The utility of visiting a node is the additional area covered by the circle, given what has been visited already. The utility function in this setting is therefore a monotone submodular set function.

In Algorithm 1 of the reference work for NAIVE, at each iteration, the non-adaptive PSPIEL_{OR} method plans a path $\mathcal{P}_t$ on the graph $\mathcal{G}$ from the current location to the goal within the remaining budget. There is no sensing because the AIPP problem has no notion of multimodal sensors. Since AIPPMS has the same state space as AIPP, we can compute $\mathcal{P}_t$ using PSPIEL_{OR} at each iteration in modified NAIVE as well. We compute the expected utility of $\mathcal{P}_t$ as follows

$$U(\mathcal{P}_t) = \sum_{v \in \mathcal{P}_t} \mathbb{E}_{b_t(\mathcal{X}_\mathcal{V})}[\Delta_F(v \mid \xi_t, \mathcal{X}_\mathcal{V})] \tag{6}$$

which is an efficient approximation of the true expected utility of the path given the current belief state $b_t(\mathcal{X}_\mathcal{V})$ and set of visited nodes ξ_t. In addition, we compute the best expected information gain among the feasible sensing actions,

$$U^*_\mathcal{S}(b_t) = \max_{\sigma \in \bar{\mathcal{A}}(s_t)} \mathrm{IG}(\sigma \mid b_t(\mathcal{X}_\mathcal{V})) \tag{7}$$

where σ^*_t is the corresponding sensor. Finally, we choose the next action to take by comparing two scaled utilities,

$$a_t = \begin{cases} \mathcal{P}_t[2] & \text{if } \lambda \cdot U(\mathcal{P}_t) > (1-\lambda) \cdot U^*_\mathcal{S}(b_t) \\ \sigma^*_t & \text{otherwise} \end{cases} \tag{8}$$

where $\lambda \in [0, 1]$ is a tuning parameter that prioritizes exploration through sensing when close to 1 and exploitation by moving when close to 0. For an ablation study of the benefit of the GCB rollout, we will also compare against POMCP using a random policy to choose actions during rollout.

5.2 Domain 1: Search-and-Rescue

Our first experimental domain is based on the search-and-rescue scenario that was presented with the original NAIVE algorithm. In the aftermath of a disaster, survivors are scattered uniformly across some terrain. The agent is an aerial vehicle with limited energy that can visit a number of locations and rescue however many people are stranded there (depending on the accessibility of a particular location to the survivors).

Table 1: Both POMCP variants significantly outperform NAIVE on three different utility distributions in the environment. The GCB Rollout is consistently better than random, though the relative gap is not very large. Rewards were averaged over 30 different trials for each setting.

Hi, Med, Lo Distribution	NAIVE	POMCP Random	POMCP GCB
$[1/6, 1/6, 2/3]$	295.7	480.0	**555.3**
$[1/3, 1/3, 1/3]$	628.3	993.0	**1020.0**
$[2/3, 1/6, 1/6]$	1104.7	1341.3	**1509.0**

The environment is represented as a unit grid, on which a location graph is generated by sampling points as nodes and joining two nodes with an edge if their mutual distance is less than some threshold ρ. The cost of traversing an edge is the scaled Euclidean distance between the two endpoint nodes. Each node or location is *independently* assigned a true state that dictates the utility of visiting it. The independence assumption is not one made by our POMCP-based approach, but allows a fairer comparison with modified NAIVE, which assumes locality (implied by independence).

For our search-and-rescue setting, there are three types of true states corresponding to low, medium, and high accessibility of the locations to survivors, represented by the area covered by the location (see Figure 4). The marginal utility of a location in a particular state, given the locations visited already, is calculated by discretizing the grid and computing the union of tiles covered by the location in that state. The key difference in our environment compared to that for NAIVE is the existence of multiple sensor models, which offer noisy observations of the true states at various locations. The sensors are modeled by a maximum fidelity parameter A, fidelity decay rate r, and energy cost C. Each usage of sensor σ incurs cost $C(\sigma)$ and yields a correct observation of the underlying state of a node with probability $P(o_i = s_i \mid s_i, \sigma) = A_\sigma \cdot r^d_\sigma$ where d is the distance between the agent and the node. Other incorrect observations have uniform likelihood given an incorrect sensor reading. Thus, $P(o_i \neq s_i \mid s_i) = (1 - A_\sigma \cdot r^d_\sigma)/2$.

Results We compared our modified POMCP (with GCB Rollout) to POMCP with Random Rollout and modified NAIVE on problems randomly generated according to the search-and-rescue scenario described above. For each problem, the location graph $\mathcal{G}$ had 30 nodes and the radius threshold ρ for edges was sampled from $\mathrm{U}(0.25, 0.4)$. We ensured via rejection sampling that each generated graph had exactly one connected component. We randomly sampled a node from $\mathcal{G}$ and set it to be both the start v_s and goal v_g. Therefore, the agent has to return to its starting point after gathering information. For each problem, we set the budget to approximately two-thirds of the Traveling Salesman cost on the graph from the start node, thereby ensuring that not all nodes could be visited and incentivizing at least some sensing.

We considered three different sets of problems, based on the distribution of accessibility types (high, medium, low) that node states are independently sampled from. For each setting, we generated 30 different problems and ran all approaches on them. Table 1 depicts the average utility or reward obtained by the approaches. All algorithms are guaranteed to be feasible by construction, so we only focus on the utility gathered. *The magnitude of the average utility is a function of the grid discretization; the relative performance is truly of interest*.

Over the problem sets, our modified POMCP with GCB Rollout consistently accrues the most utility, significantly outperforming modified NAIVE and also POMCP with Random Rollout, albeit to a lesser extent (the relative performance gap between GCB and Random Rollout increases on a more challenging problem in Section 5.3). As expected, with an increase in the proportion of the higher utility states, the absolute utility accrued by all approaches increases. More notably, the *relative performance gap is highest for the first set*, where high utility states are least prevalent. This suggests that the modified POMCP balances sensing and movement more effectively than modified NAIVE when identifying which states are of likely higher or lower utility. The average iteration of NAIVE requires 1 s of computation and that of POMCP requires 6 s. Both are reasonable for our purposes, and increasing the computation time for NAIVE through the relevant parameters did not improve performance.

Fundamental differences between a POMCP approach and NAIVE explain the performance gap on AIPPMS, over and above implementation quality. Each candidate path computed by modified NAIVE ignores sensing actions. Subsequently, NAIVE computes the expected information gain of possible sensing actions from the current belief state and then compares that with the candidate path to decide whether to move or sense. However, the lookahead search in POMCP can simulate the effects of movement followed by sensing, and sensing followed by movement. Therefore, it can identify some good future sequences of potentially *interleaved sensing and movement actions*, and then decide which is the next best action to take.

5.3 Domain 2: Information Search RockSample

To further motivate the AIPPMS formulation and evaluate our approach, we also adapt the Information Search RockSample domain or ISRS (He, Brunskill, and Roy 2011). It is a variant of the classical RockSample problem for POMDPs that is both far more challenging and more suitable for comparing informative path planning algorithms. Briefly, the ISRS problem models a rover exploring an unknown terrain, represented as an $n \times n$ grid. Scattered over the grid are k rocks, with at most one rock per grid cell. Only some of the rocks are 'good', i.e. have scientific value and yield a positive reward. Once a rock is visited it becomes 'bad' and provides no further reward when sampled. The positions of the rover and rocks are known apriori, but only visiting a rock reveals its state. See Figure 5 for an illustration.

There is also a set of b beacons (one per cell, no overlap with rock locations) in the grid. The rover *must visit the beacons in order to take sensing actions* and get observations about the state of the nearby rocks, where the fidelity of the observation reduces with increasing distance between the beacon and the corresponding rock. As before, there are multiple sensor modalities with complementary trade-offs of usage energy and fidelity parameters. Moving between adjacent cells also expends energy cost. The rover must return to the origin cell without exceeding its energy budget.

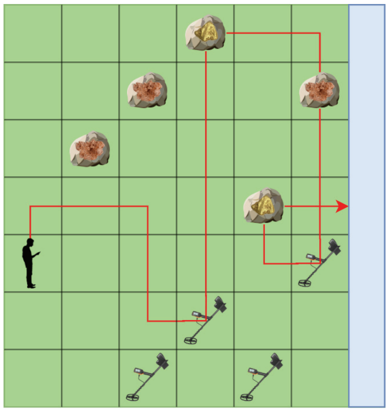

Figure 5: An illustration of the Information Search RockSample problem, our second evaluation domain. The agent uses beacons to sense the states of rocks, and gathers utility by visiting good rocks.

Results For ISRS, we focus on the interplay between three problem parameters: the number of rocks k, the number of beacons b, and the relative proportion of good rocks, through the independent Bernoulli probability p of a rock being good. Accordingly, we vary these three parameters for our experiments while keeping the others fixed. We set the size of the grid to 10×10, the energy budget to be 100 units, where the energy cost of each movement is 1 unit and that of using the two sensors are 0.5 and 2 units respectively, and the reward for sampling a good rock to be 10 units.

Table 2 compares the average reward for POMCP with the random rollout strategy to that for POMCP with our GCB rollout strategy, over a range of $\{k, b, p\}$ settings, averaged over 30 trials for each setting. NAIVE accrued little to no reward for most settings and trials, so we omitted it in the interest of space. This behavior of modified NAIVE is not too surprising. For the ISRS domain with its beacons, explicitly reasoning about the trade-off between movement and multimodal sensing is particularly important for good performance. *Therefore, the ISRS domain provides strong empirical justification for extending the AIPP formulation to incorporate multimodal sensing, in addition to our earlier justification from first principles.*

We highlight **three key observations** from the readings in Table 2. First and foremost, GCB consistently outperforms Random across all settings, far more so than it did for the search-and-rescue domain. This finding further underscores how ISRS is significantly more challenging than search-and-rescue. Second, the relative performance gap of GCB to Random increases both with more good rocks (higher k) and with a higher proportion of good rocks (higher p), e.g. compare the relative performance for $\{10, 10, 0.5\}$ to both

Table 2: On the Information Search Rock Sample domain, the GCB rollout significantly outperforms the random rollout over all settings by making better use of beacons and visiting more good rocks. We averaged the rewards from 50 different trials for each setting. The standard error of the mean is less than 10% of the mean in each case.

Rocks	Beacons	$p = 0.5$		$p = 0.75$		$p = 1.0$	
k	b	Random	GCB	Random	GCB	Random	GCB
10	10	21.6	**29.4**	25.2	**38.2**	30.0	**49.0**
10	25	23.4	**27.8**	26.8	**41.0**	24.2	**47.4**
25	10	45.0	**63.6**	54.2	**87.8**	63.6	**121.8**
25	25	41.8	**77.0**	52.6	**105.0**	69.4	**120.8**

$\{10, 10, 0.75\}$ and to $\{25, 25, 0.5\}$. Third, for the same k (rocks) but with increasing b (beacons), the performance of GCB relative to itself either stays the same or increases, e.g., compare GCB for $\{25, 10, 0.75\}$ to $\{25, 25, 0.75\}$, while for Random it does not increase appreciably for any setting. This supports the intuitive hypothesis that GCB is making better use of environmental information to improve its estimate of the goodness of a rock.

6 Conclusion

We extended the Adaptive Informative Path Planning problem to a setting with Multimodal Sensing (AIPPMS). In contrast to previous AIPP approaches that eschew a POMDP formulation as being too general and intractable, we embraced POMDPs as the appropriate structure to jointly reason about movement and sensing for the more general AIPPMS problem. Due to the large state and observation space and implicit reachability structure of AIPPMS, we used the online planning framework of Partially Observable Monte Carlo Planning, with modifications to the action selection and an adaptive greedy rollout policy based on Generalized Cost-Benefit. Our resulting approach consistently outperforms the modified NAIVE algorithm over multiple domains, and our rollout policy is a key contributor to this performance.

Future research could address many of our limitations. We take an empirical approach in this paper in contrast to most work on AIPP that conducts theoretical analyses (albeit under modeling assumptions). A more rigorous approach to analyzing AIPPMS, under appropriate assumptions on the utility and sensor models, would be of interest and may precipitate the development of high-performance tailored algorithms. As we motivated earlier, we used the POMCP framework for its simplicity and ease of extension, but an extensive study of other state-of-the-art solvers, both offline and online, would be instructive. In our AIPPMS formulation, the inter-dependency between movement and sensing actions is weak; the costs and utilities of the actions of each type are unaffected by actions of the other type. Perhaps the POMDP approach would have even more substantial gains in performance if the coupling between action types was stronger (this would require different utility and cost models). Finally, given the greater applicability of AIPPMS over AIPP to real-world robotics problems with energy costs for sensing, implementing the modified POMCP for such an application would be appropriate.

References

Bezanson, J.; Edelman, A.; Karpinski, S.; and Shah, V. 2017. Julia: A Fresh Approach to Numerical Computing. *SIAM Review* 59(1):65–98.

Binney, J., and Sukhatme, G. S. 2012. Branch and Bound for Informative Path Planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2147–2154.

Binney, J.; Krause, A.; and Sukhatme, G. S. 2010. Informative Path Planning for an Autonomous Underwater Vehicle. In *IEEE International Conference on Robotics and Automation (ICRA)*, 4791–4796.

Canny, J. F. 1987. The Complexity of Robot Motion Planning. *ACM Doctoral Dissertation Award*.

Charrow, B.; Kahn, G.; Patil, S.; Liu, S.; Goldberg, K.; Abbeel, P.; Michael, N.; and Kumar, V. 2015. Information-Theoretic Planning with Trajectory Optimization for Dense 3D Mapping. In *Robotics: Science and Systems*.

Chen, W., and Liu, L. 2019. Pareto Monte Carlo Tree Search for Multi-Objective Informative Planning. In *Robotics: Science and Systems*.

Choudhury, S.; Kapoor, A.; Ranade, G.; and Dey, D. 2017. Learning to Gather Information via Imitation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 908–915.

Choudhury, S.; Bhardwaj, M.; Arora, S.; Kapoor, A.; Ranade, G.; Scherer, S.; and Dey, D. 2018. Data-driven Planning via Imitation Learning. *International Journal of Robotics Research* 37(13-14):1632–1672.

Das, J.; Py, F.; Harvey, J. B.; Ryan, J. P.; Gellene, A.; Graham, R.; Caron, D. A.; Rajan, K.; Sukhatme, G. S.; et al. 2015. Data-driven Robotic Sampling for Marine Ecosystem Monitoring. *International Journal of Robotics Research* 34(12):1435–1452.

Dressel, L., and Kochenderfer, M. J. 2017. Efficient Decision-Theoretic Target Localization. In *International Conference on Automated Planning and Scheduling (ICAPS)*.

Egorov, M.; Sunberg, Z. N.; Balaban, E.; Wheeler, T. A.; Gupta, J. K.; and Kochenderfer, M. J. 2017. Pomdps. jl: A Framework for Sequential Decision Making under Uncertainty. *Journal of Machine Learning Research* 18(26):1–5.

Floyd, R. W. 1962. Algorithm 97: Shortest Path. *Communications of the ACM* 5(6):345.

Golden, B. L.; Levy, L.; and Vohra, R. 1987. The Orienteering Problem. *Naval Research Logistics (NRL)* 34(3):307–318.

Golovin, D., and Krause, A. 2011. Adaptive Submodularity: Theory and Applications in Active Learning and Stochastic Optimization. *Journal of Artificial Intelligence Research* 42:427–486.

Hauskrecht, M. 2000. Value-function Approximations for Partially Observable Markov Decision Processes. *Journal of Artificial Intelligence Research* 13:33–94.

He, R.; Brunskill, E.; and Roy, N. 2011. Efficient Planning under Uncertainty with Macro-actions. *Journal of Artificial Intelligence Research* 40:523–570.

Heng, L.; Gotovos, A.; Krause, A.; and Pollefeys, M. 2015. Efficient Visual Exploration and Coverage with a Micro Aerial Vehicle in Unknown Environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, 1071–1078.

Hollinger, G. A., and Sukhatme, G. S. 2013. Sampling-based Motion Planning for Robotic Information Gathering. In *Robotics: Science and Systems*, volume 3.

Kochenderfer, M. J. 2015. *Decision Making under Uncertainty: Theory and Application*. MIT Press.

Krause, A., and Golovin, D. 2014. Submodular Function Maximization. In *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press. 71–104.

Lim, Z. W.; Hsu, D.; and Lee, W. S. 2016. Adaptive Informative Path Planning in Metric Spaces. *International Journal of Robotics Research* 35(5):585–598.

Meliou, A.; Krause, A.; Guestrin, C.; and Hellerstein, J. M. 2007. Nonmyopic Informative Path Planning in Spatio-temporal Models. In *AAAI Conference on Artificial Intelligence (AAAI)*, 602–607.

Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An Analysis of Approximations for Maximizing Submodular Set Functions-I. *Mathematical Programming* 14(1):265–294.

Papadimitriou, C. H., and Tsitsiklis, J. N. 1987. The Complexity of Markov Decision Processes. *Mathematics of Operations Research* 12(3):441–450.

Ross, S.; Pineau, J.; Paquet, S.; and Chaib-Draa, B. 2008. Online Planning Algorithms for POMDPs. *Journal of Artificial Intelligence Research* 32:663–704.

Shimony, S. E. 1994. Finding MAPs for Belief Networks is NP-hard. *Artificial Intelligence* 68(2):399–410.

Silver, D., and Veness, J. 2010. Monte-Carlo Planning in Large POMDPs. In *Advances in Neural Information Processing Systems (NIPS)*, 2164–2172.

Singh, A.; Krause, A.; Guestrin, C.; Kaiser, W. J.; and Batalin, M. A. 2007. Efficient Planning of Informative Paths for Multiple Robots. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 7, 2204–2211.

Singh, A.; Krause, A.; Guestrin, C.; and Kaiser, W. J. 2009. Efficient Informative Sensing using Multiple Robots. *Journal of Artificial Intelligence Research* 34:707–755.

Singh, A.; Krause, A.; and Kaiser, W. J. 2009. Nonmyopic Adaptive Informative Path Planning for Multiple Robots. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Smallwood, R. D., and Sondik, E. J. 1973. The Optimal Control of Partially Observable Markov Processes over a Finite Horizon. *Operations Research* 21(5).

Sondik, E. J. 1978. The Optimal Control of Partially Observable Markov Processes over the Infinite Horizon: Discounted Costs. *Operations Research* 26(2).

Stachniss, C.; Grisetti, G.; and Burgard, W. 2005. Information Gain-based Exploration using Rao-Blackwellized Particle Filters. In *Robotics: Science and Systems*, volume 2, 65–72.

Sunberg, Z. N., and Kochenderfer, M. J. 2018. Online Algorithms for POMDPs with Continuous State, Action, and Observation Spaces. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 259–263.

Ye, N.; Somani, A.; Hsu, D.; and Lee, W. S. 2017. Despot: Online POMDP Planning with Regularization. *Journal of Artificial Intelligence Research* 58:231–266.

Zhang, H., and Vorobeychik, Y. 2016. Submodular Optimization with Routing Constraints. In *AAAI Conference on Artificial Intelligence (AAAI)*, 819–825.