Approximate sequential optimization for informative path planning

Joshua Ott [a],*, Mykel J. Kochenderfer [a], Stephen Boyd [b]

[a] *Department of Aeronautics & Astronautics, Stanford University, USA*
[b] *Department of Electrical Engineering, Stanford University, USA*

ARTICLE INFO

ABSTRACT

We consider the problem of finding an informative path through a graph, given initial and terminal nodes and a given maximum path length. We assume that a linear noise corrupted measurement is taken at each node of an underlying unknown vector that we wish to estimate. The informativeness is measured by the reduction in uncertainty in our estimate, evaluated using several Gaussian process-based metrics. We present a convex relaxation for this informative path planning problem, which we can readily solve to obtain a bound on the possible performance. We develop an approximate sequential method where the path is constructed segment by segment through dynamic programming. This involves solving an orienteering problem, with the node reward acting as a surrogate for informativeness, taking the first step, and then repeating the process. The method scales to very large problem instances and achieves performance close to the bound produced by the convex relaxation. We also demonstrate our method's ability to handle adaptive objectives, multimodal sensing, and multi-agent variations of the informative path planning problem.

1. Introduction

Consider an agent that is tasked with exploring a large unknown environment. The agent must collect measurements to build an accurate representation of the state of the environment. The agent has a finite set of resources and is therefore constrained by time, battery life, or fuel capacity. As a result, it must plan a path to maximize the amount of information acquired from its noisy observations while remaining within the allowed resource budget. Given a maximum path length L^{max}, the objective is to compute a path of length at most L^{max} that maximizes the gain in information over the prediction variables. This is known as the informative path planning (IPP) problem and involves choosing a subset of sensing locations from the environment. The IPP problem can be seen as an extension of the sensor selection problem where future sensing locations are constrained by previous sensing locations. These additional path feasibility constraints increase the complexity of the problem, making it more challenging than the original sensor selection problem [1,2].

A variety of real-world problems can be formulated as IPP problems, such as planetary rover exploration, search and rescue, and environmental monitoring. IPP merges the fields of robotics, artificial intelligence, and spatial data analysis. It carries substantial implications for a myriad of applications, where efficient data collection from a potentially hazardous or inaccessible environment is essential. However, the problem presents significant challenges. The IPP problem is known to be NP-hard due to its combinatorial nature with a vast search space for potential solutions [3]. Successful solutions typically require reasoning about the entire trajectory; myopic approaches often result in inefficient paths, causing the agent to repeat exploration of the same areas without using the full resource budget [4–8].

The IPP problem is closely related to the optimal design of experiments and sensor selection. Joshi and Boyd presented a convex relaxation of the sensor selection problem that involves choosing a set of k sensor measurements, from a set of m possible or potential sensor measurements, that minimizes the error in estimating some parameters [1]. The sensor selection problem is closely related to the IPP problem. However, sensor selection does not consider the movement constraints of the agent so the rounding approach presented by Joshi and Boyd is not directly applicable to the IPP problem. Krause et al. also considered the sensor placement problem and presented several approximate solution methods [9]. Singh et al. introduced an approximate algorithm for both the single and multi-agent IPP problems using Gaussian processes as the underlying environment model, but they do not consider the adaptive IPP problem [10]. Strawser and Williams used hybrid search to solve risk-bound goal-directed planning problems, focusing on complex 3D environments and the modeling of trajectory risk [11]. Bäckström et al. focused on approximability for cost-optimal planning [12].

Dutta et al. introduced a mixed-integer formulation for the discrete IPP problem in random fields and aimed to optimize over the collected measurement subset and all linear estimators [2]. The approach

---

* Corresponding author.

E-mail addresses: joshuaott@stanford.edu (J. Ott), mykel@stanford.edu (M.J. Kochenderfer), boyd@stanford.edu (S. Boyd).

presented by Dutta et al. can produce globally optimal solutions on relatively small graphs; however, the mixed integer program fails to scale to larger graph sizes which is often necessary in practice for planning at higher resolutions [2]. Boström-Rost et al. formulated the problem as a mixed integer semidefinite program and used a receding horizon optimal control approach based on the information filter to produce a solution [13]. Song et al. used a receding horizon control strategy with a discrete set of motion primitives to evaluate the objective and then select the best candidate, execute the action, update the Gaussian process belief, and then replan [14].

In addition to the discrete path planning methods discussed, continuous informative path planning approaches have also been explored. These methods focus on optimizing continuous paths rather than discrete sensing locations, which is crucial for certain real-world applications where smooth trajectories and compliance with motion constraints are required. Hitz et al. used an evolutionary strategy to optimize a parameterized path in continuous space [15]. Francis et al. presented a constrained Bayesian optimization approach to plan continuous paths that inherently satisfy motion and safety constraints while balancing the reward and risk associated with each path [16]. Asgharivaskasi et al. proposed a differentiable approximation of the Shannon mutual information between a grid map and ray-based observations, enabling gradient ascent optimization in the continuous space of SE(3) sensor poses [17].

Another common approach is to formulate the IPP problem as a partially observable Markov decision process (POMDP). Marchant et al. formulated the IPP problem as a POMDP and solved it using sequential Bayesian optimization through Monte Carlo tree search with upper confidence bound for trees (MCTS-UCT) [18]. This work has been extended to modify the reward function for achieving monitoring behavior that exploits areas with high gradients, and for additional reasoning over continuous action spaces through Bayesian optimization [4,19]. Fernández et al., on the other hand, used partially observable Monte Carlo planning (POMCP) with Gaussian process beliefs for estimating quantiles of the underlying world state [20]. Their methodology proposes sample locations for a team of scientists to investigate following their exploration completion. Rückin et al. presented a hybrid approach that combines tree search with an offline-learned neural network to predict informative sensing actions [21]. Ott et al. focused on the multimodal sensing extension of the IPP problem where the agent has multiple sensors to choose from and solved it with tree search [22].

These approaches all produce reasonable solutions on graphs with a relatively small number of nodes; however, they struggle to scale to large graphs with higher resolution which is necessary for many real-world applications. Additionally, MIPs are not well-suited to handle adaptive objectives where the objective function depends on the history of measurement values received. Our results demonstrate that our approach is able to scale to larger problem instances for the IPP problem, while also handling adaptive objectives such as the expected improvement metric, and accounting for multimodal sensing and multi-agent variations of the IPP problem [22,23]. We directly compare our approach with those of Dutta et al. and Ott et al. [2,22].

The key contributions of this work are the following:

1. We present a convex relaxation of the informative path planning problem.
2. We introduce an approximate sequential optimization solution that outperforms existing IPP methods.
3. We provide a bound on the optimal solution to the IPP problem and show that while our method produces a suboptimal solution, often the optimality gap is small.
4. We extend our method to handle adaptive objectives such as the expected improvement metric, as well as other variations of the IPP problem including multimodal sensing and multi-agent scenarios [24].

5. We release our implementation as an open-source software package for use and further development by the community.[1]

The remainder of this paper is organized as follows: Section 2 presents the IPP problem, Section 3 formulates the IPP problem with an exact mixed integer convex formulation and presents a convex relaxation, Section 4 introduces our approximate sequential path optimization solution, Section 5 extends our method to consider additional adaptive objectives and multimodal sensing variations of the original problem, Section 6 presents our empirical results along with a bound on the optimality of our approximate solution, and Section 7 concludes our work.

2. Informative path planning

2.1. Feasible paths

We model the environment as a directed graph with vertices $V = \{1, \ldots, n\}$ and edges $\mathcal{E} \subseteq V \times V$. Our goal is to find a feasible path p from the start vertex s to the goal vertex g where we assume that such a feasible path exists. We denote a path of length L as $p = (i_1, i_2, \ldots, i_L)$ where $i_1 = s$ and $i_L = g$. A feasible path is one that starts at s and ends at g while remaining within the given maximum path length $L^{\max}$; that is, $L \leq L^{\max}$.

In practice, the vertices of the graph correspond to the locations in the environment we are interested in exploring. Additionally, environmental obstacles can be incorporated by removing edges in the graph. However, these practical considerations do not alter the generality of our formulation.

2.2. Sensing along a path

We are interested in estimating a vector $x \in \mathbf{R}^m$ from a set of linear measurements, corrupted by additive noise,

$$y_i = a_i^T x + v_i \tag{1}$$

where $x \in \mathbf{R}^m$ is a vector of parameters to estimate, each $v_i \sim \mathcal{N}(0, \sigma^2)$ is independent and identically distributed additive noise, and $a_i \in \mathbf{R}^m$ characterizes the measurements.

We assume we have some prior over x with known mean and covariance $\mathcal{N}(\bar{x}, \Sigma_x)$. Here, x can represent subsurface temperature, ice concentration, algae density, or a variety of other spatially varying phenomena.

We assume that at each node i_k along a path p we make measurement y_k. The measurements received along a path p of length L are then given by

$$y_k = a_{i_k}^T x + v_{i_k} \quad k = 1, \ldots, L. \tag{2}$$

The maximum a posteriori estimate of x is then

$$\hat{x} = \bar{x} + \left(\sigma^{-2} \sum_{k=1}^{L} a_{i_k} a_{i_k}^T + \Sigma_x^{-1} \right)^{-1} \sigma^{-2} \sum_{k=1}^{L} (y_k - \bar{y}_k) a_{i_k} \tag{3}$$

where $\hat{x} \in \mathbf{R}^m$ with covariance

$$\Sigma = \left(\sigma^{-2} \sum_{k=1}^{L} a_{i_k} a_{i_k}^T + \Sigma_x^{-1} \right)^{-1} \tag{4}$$

where $\bar{x}$ is the prior mean and $\bar{y}_k = a_{i_k}^T \bar{x}$. Note that this is exactly the Gaussian process mean and covariance as shown in Appendix D.

2.3. Measures of informativeness

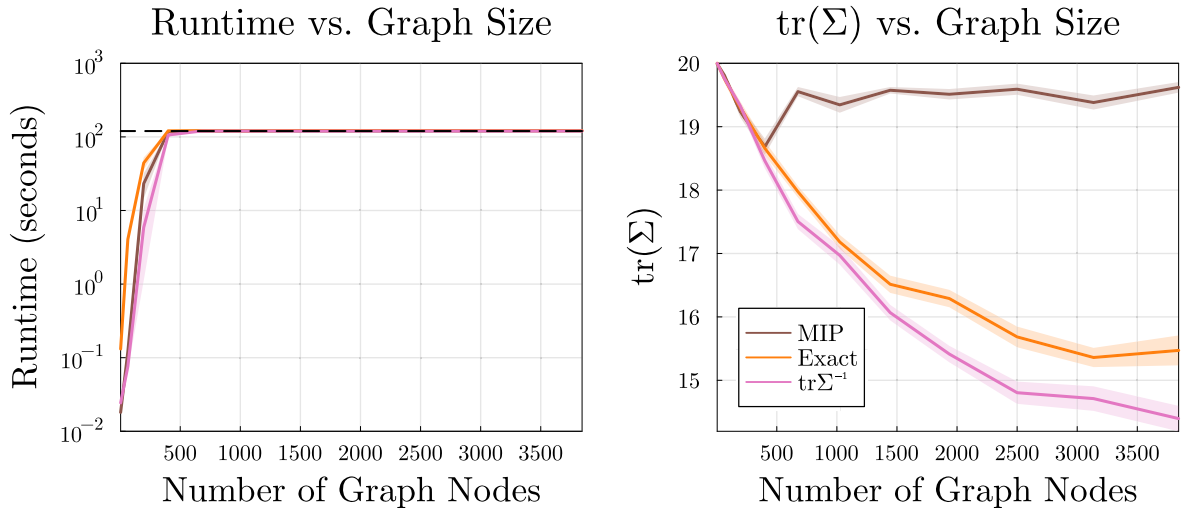Three common scalar measures of the informativeness of a path are:

Fig. 1. Comparative analysis of computational runtime (left) and the objective function $\mathbf{tr}(\Sigma)$ (right) as functions of the graph size. Three methods are compared: the mixed integer program formulation [2], the exact formulation from Eq. (16), and the exact method using the B-IPP $\mathbf{tr}(\Sigma^{-1})$ objective. Each curve shows the average runtime and objective value respectively with the standard error reported over 25 different simulations. The dashed line on the left indicates the 120 s runtime constraint imposed on all methods.

1. $\mathbf{tr}(\Sigma) = \mathbf{E}\left(\|x - \hat{x}\|^2\right)$, which corresponds to the mean squared error in estimating the parameter x.

2. $-\mathbf{tr}(\Sigma^{-1})$, which is the negative trace of the information matrix. We include the negative here since we are interested in minimizing each of these objectives in order to maximize the informativeness of the path.

3. $\mathrm{logdet}(\Sigma)$, which is proportional to the log volume (or mean radius) of the resulting confidence ellipsoid.

Note that each scalar measure of informativeness is a function of the eigenvalues of Σ. We will denote the measure of informativeness of a path as $\phi(p)$ where smaller values of $\phi(p)$ correspond to more informative paths. Other measures of informativeness such as adaptive objectives will be discussed in Section 5.2 and we also provide a formulation of the mutual information in Appendix E.

2.4. Informative path planning problem

The informative path planning (IPP) problem seeks to find the maximally informative feasible path p from the set of all feasible paths $\mathcal{P}$. The IPP problem can be stated as follows

$$\begin{aligned} \text{minimize} \quad & \phi(p) \\ \text{subject to} \quad & p \in \mathcal{P} \end{aligned} \tag{5}$$

where the variables are the nodes in the path p and the data consists of the graph, the measurement characterization a_i, the measurement variance σ^2, and the prior covariance Σ_x. This is a combinatorial problem since the variable is a path.

2.5. Relation to A and D-optimal experiment design

Depending on the information objective used, the informative path planning problem is closely related to the experiment design problem. The experiment design problem involves choosing c out of h available sensors. For each of the selected c sensors, we also need to determine how many times it should be used while respecting the constraints on the number of total allowable uses [1]. A-optimal experiment design seeks to minimize $\mathbf{tr}(\Sigma)$ as the information objective while D-optimal design seeks to minimize $\mathrm{logdet}(\Sigma)$. While not commonly named, we introduce B-optimal design which seeks to minimize $-\mathbf{tr}(\Sigma^{-1})$ and we show in Section 6.2 that it proves to be a powerful surrogate for both A and D-optimal objectives.

Informative path planning focuses on selecting specific sensing locations from the entire environment. While it builds upon the sensor selection problem, it introduces additional constraints: future sensing locations can only be chosen if they are adjacent to previously selected locations. These path feasibility constraints introduce additional combinatorial complexity, making the problem more challenging than the original sensor selection task.

We will refer to the $\mathbf{tr}(\Sigma)$, $\mathrm{logdet}(\Sigma)$, $-\mathbf{tr}(\Sigma^{-1})$ objectives as the A-IPP, D-IPP, and B-IPP objectives respectively. For the A-IPP objective we seek to minimize $\phi(p) = \mathbf{tr}(\Sigma)$, for the B-IPP objective we seek to minimize $\phi(p) = -\mathbf{tr}(\Sigma^{-1})$, and for the D-IPP objective we seek to minimize $\phi(p) = \mathrm{logdet}(\Sigma)$.

3. Mixed integer convex formulation

We now introduce a Boolean matrix $z \in \mathbf{R}^{n \times n}$ which we will use to parameterize a path p. Here, $z_{ij} = 1$ if the edge from node i to node j is in the path p. We must have,

$$z_{ij} = 0 \quad \text{for} \quad (i, j) \notin \mathcal{E}. \tag{6}$$

That is, the sparsity pattern of z is the same as that of the environment graph.

Using the Boolean parametrization of z we can define a feasible path. A feasible path must remain within the given maximum path length $L^{\max}$:

$$\sum_{i=1}^{n-1} \sum_{j \in N_{out}(i)} z_{ij} \leq L^{\max} \tag{7}$$

where $N_{out}(i)$ denotes the graph neighbors leaving node i and $N_{in}(i)$ denotes the graph neighbors entering node i.

If we take node 1 and n as the start and goal nodes respectively, then we know the start and end nodes must exist on the path meaning that one edge going out of the start node and one edge going into the goal node must be in the path:

$$\sum_{i \in N_{out}(1)} z_{1i} = \sum_{j \in N_{in}(n)} z_{jn} = 1. \tag{8}$$

Similarly, to enforce the start and termination of the path, we do not allow any edges to enter the starting node or leave the goal node:

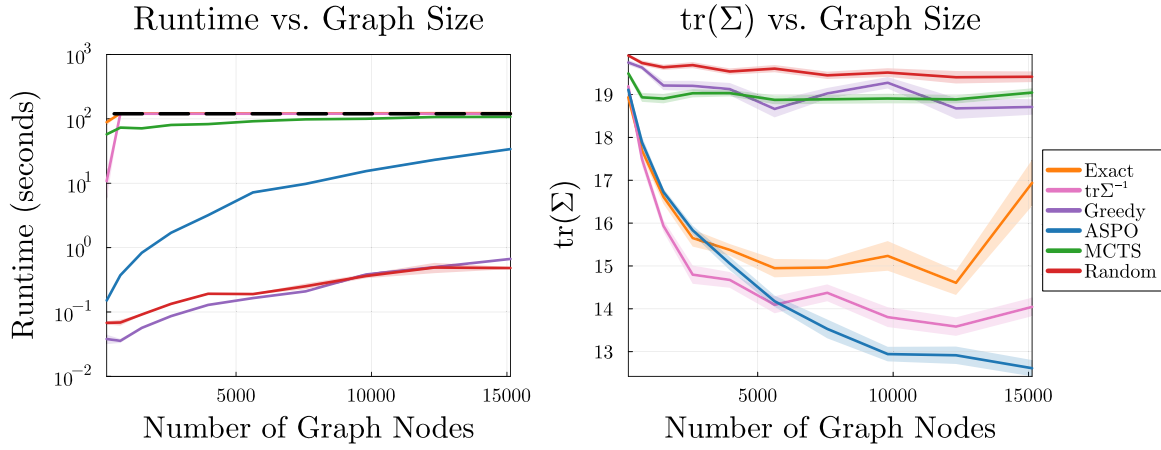$$\sum_{i \in N_{in}(1)} z_{i1} = \sum_{j \in N_{out}(n)} z_{nj} = 0. \tag{9}$$

Fig. 2. Runtime and A-IPP objective as a function of graph size for the six methods considered. Each curve shows the average runtime and objective value respectively with the standard error reported over 25 different simulations. The dashed line on the left indicates the 120 second runtime constraint imposed on all methods.

To ensure the connectivity of the path and that every node is visited at most once, we enforce

$$\sum_{\substack{j \in N_{out}(i) \\ j \neq 1}} z_{ij} = \sum_{\substack{k \in N_{in}(i) \\ k \neq n}} z_{ki} \leq 1 \quad i = 2, \ldots, n-1. \quad (10)$$

To prevent any subtours from occurring, we use the Miller-Tucker-Zemlin constraints for subtour elimination [25]. To do so, we introduce the variable u which enforces that node j must follow node i if edge (i, j) is in the path p:

$$u_i - u_j + 1 \leq (n-1)(1 - z_{ij}) \quad 2 \leq i \neq j \leq n \quad (11)$$

$$2 \leq u_i \leq n \quad 2 \leq i \leq n \quad (12)$$

$$u_1 = 1. \quad (13)$$

Finally, to enforce the integrality of the Boolean matrix z we have:

$$z_{ij} \in \{0, 1\} \quad i = 1, \ldots, n. \quad (14)$$

The constraints provided in Eqs. (7)–(14) fully define the space of feasible path parameterizations $\mathcal{Z}$. We then have

$$\Sigma_z = \left(\sigma^{-2} \sum_{i=1}^n \left(\sum_{j \in N_{out}(i)} z_{ij} \right) a_i a_i^T + \Sigma_x^{-1} \right)^{-1} \quad (15)$$

and can write our parameterized objective as $\phi(z)$. We then have the following convex mixed integer problem:

$$\begin{aligned} \text{minimize} \quad & \phi(z) \\ \text{subject to} \quad & z \in \mathcal{Z}. \end{aligned} \quad (16)$$

While it is not immediately obvious, the $\phi(z)$ objective is convex in both the A, B, and D-IPP objectives introduced in Section 2.3 as discussed in section 4.6.2 of Boyd and Vandenberghe. The problem would be entirely convex if it were not for the integer constraints on z in Eq. (14).

3.1. Convex relaxation

The exact mixed integer convex problem in Eq. (16) can be converted to a fully convex problem by relaxing the integer constraints on z. Specifically, we replace the $z_{ij} \in \{0, 1\}$ constraint with $0 \leq z_{ij} \leq 1$.

The solution to the relaxed problem is denoted as $\bar{z}$ and provides a lower bound ℓ on the optimal solution $z^\star$. The reason for this is that the feasible set for the relaxed problem contains the feasible set for the original problem and therefore the optimal value of the original problem cannot be smaller than that of the relaxed problem [26]. In the case where $\bar{z}$ is Boolean, then we can say that $z^\star = \bar{z}$.

4. Approximate sequential path optimization

The exact formulation provided in Section 3 is a mixed-integer convex problem (MICP). For relatively small problems, these MICPs are solvable, and we provide an implementation using the Pajarito solver in Julia.[2] However, the MICPs become prohibitively expensive computationally as n grows. The relaxed version of the problem is able to scale to much larger problems ($n > 10000$).

We now introduce our method, which we refer to as approximate sequential path optimization (ASPO), which achieves similar performance to that of the exact mixed integer solution while maintaining a similar speed and scalability of the relaxed version.

Our approximate method begins with the agent at the starting vertex assuming no measurements have been taken yet. For each of the n vertices in the environment, we compute r_j, which is the estimated value of visiting node j conditioned on the previously visited nodes. For example,

$$r_j = -\phi((p_t, j)) \quad (17)$$

which is the objective value if node j was added to the current executed path p_t at time t. Note the negative sign since we seek to minimize the $\phi(p)$ objective which corresponds to maximizing the informativeness of a path.

The objective is to find a path from s to g that maximizes the total reward collected, subject to a constraint on the total path length not exceeding L^{max}. This is known as the Orienteering Problem (OP) [27, 28] and is formally stated as:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n-1} \sum_{j \in N_{out}(i)} z_{ij} r_j \\ \text{subject to} \quad & z \in \mathcal{Z}. \end{aligned} \quad (18)$$

Solving the exact OP is NP-hard [28]; however, we can use dynamic programming to solve an approximate version of the OP efficiently. Let $U_{i,b}$ denote the value for each node i and remaining path length b. Then we have

$$U_{i,b} = r_i + \max_{j \in N_{out}(i)} (U_{j,b-1}). \quad (19)$$

The path $\hat{p}_t$ is then constructed by starting at s and following the maximum $U_{j,b}$ value for $j \in N_{out}(i)$. The first step, or first h steps, of the candidate $\hat{p}_t$ are executed and the problem is then re-solved with the newly executed path and the updated values for r_j.
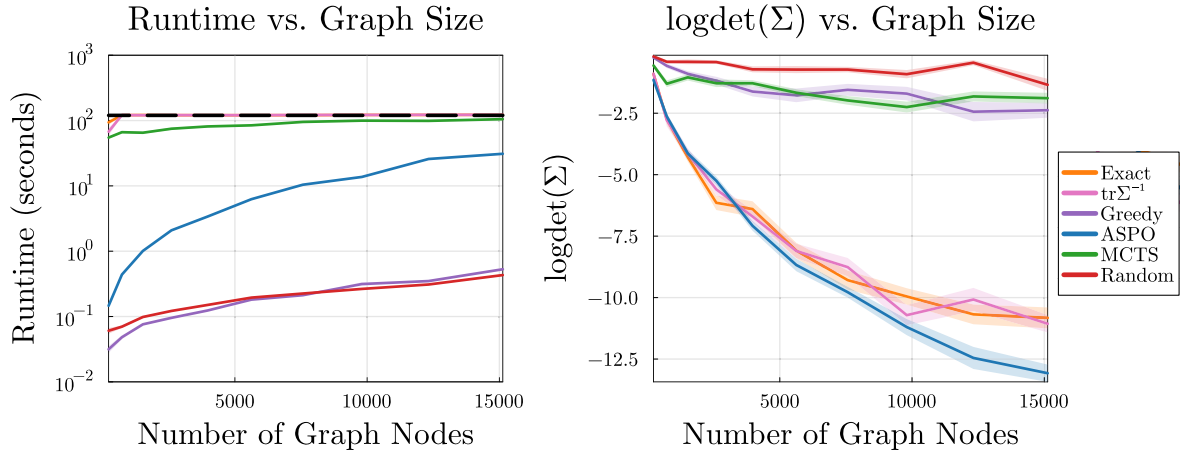

Fig. 3. Runtime and D-IPP objective as a function of graph size for the six methods considered. Each curve shows the average runtime and objective value respectively with the standard error reported over 25 different simulations. The dashed line on the left indicates the 120 second runtime constraint imposed on all methods.

Sequentially constructing the path in this receding horizon approach allows for efficient solutions to relatively large problems. The key reason for this is that by approximating the value of node j we are better able to isolate valuable regions of the environment to visit. While this assumption assumes that the agent can essentially teleport from its current location to anywhere in the reachable environment, it proves to be useful from the computational savings it provides and allows for many successive iterations of the problem to construct the path in a receding horizon fashion. As a result, the approximation of the value r_j improves as more of the path is executed. This heuristic proves to be valuable because it allows the problem to remain tractable for large n.

4.1. Optimality gap

The path p obtained from our approximate sequential path optimization can be used to construct the matrix $\hat{z}$ which gives an upper bound u on the value of the IPP problem. From Section 3, we know that the relaxed problem provides a lower bound ℓ on the optimal solution. Therefore, the difference between the upper and lower bounds gives us the optimality gap

$$\delta = \frac{1}{m}\left(u - \ell\right) = \frac{1}{m}\left(\phi(\hat{z}) - \phi(\tilde{z})\right). \tag{20}$$

For D-IPP we can express the gap as the ratio of mean radii e^δ. We know that $\delta \geq 0$. If $\delta = 0$ then $\hat{z}$ is the optimal solution for the IPP problem $\hat{z} = z^\star$. We can say that our approximate sequential path optimization solution is no more than δ-suboptimal. We present empirical results with evaluations of the optimality gap in Section 6.5.

5. Extensions and variations

The IPP problem has several extensions inspired by numerous real-world applications. We introduce three such extensions below that have been widely studied in the literature. We demonstrate the flexibility of our approximate sequential path optimization approach in handling these extensions which are not easily handled by mixed integer formulations.

5.1. Edge weights

Our formulation provided thus far has used a constraint on the maximum path length. The IPP problem can also be formulated where each edge $(i, j) \in \mathcal{E}$ is assigned a weight d_{ij}, where d_{ij} denotes the weight of the edge from vertex i to vertex j. The maximum path length constraint then becomes a budget constraint where the path cost must not exceed a given budget B. That is, for each node i_k in path p we

have $(i_k, i_{k+1}) \in \mathcal{E}$ and $\sum_{k=1}^{L-1} d_{i_k, i_{k+1}} \leq B$. The constraint in Eq. (7) is then given by

$$\sum_{i=1}^{n-1} \sum_{j \in N_{out}(i)} z_{ij} d_{ij} \leq B. \tag{21}$$

5.2. Additional objectives

The approximate sequential path optimization formulation we presented above allows for the inclusion of adaptive objectives, like lower confidence bound and expected improvement, that depend on the received sample values [24]. Incorporating adaptive objectives is typically referred to as the adaptive informative path planning (AIPP) problem [22,23]. These adaptive objectives cannot be handled by mixed integer formulations because the objectives require reacting to new observations as they become available. In other words, the belief about the true state of the environment changes in response to the values of the observations. So while A and D-optimal experiment design can be computed a priori, adaptive objectives must be reasoned about in a receding horizon fashion.

The expected improvement metric at node i is given by:

$$\mathbb{E}[I_p]_j = (y_{\min} - \hat{x}_j)\Phi\left(\frac{y_{\min} - \hat{x}_j}{\hat{\sigma}_j}\right) + \hat{\sigma}_j^2 \mathcal{N}(y_{\min} \mid \hat{x}_j, \hat{\sigma}_j^2) \tag{22}$$

where $y_{\min}$ is the current minimum value found so far, and $\Phi(\cdot)$ is the cumulative distribution function [24]. Our problem then becomes

$$\text{minimize} \quad \sum_{j=1}^{m} \mathbb{E}[I_p]_j$$
$$\text{subject to} \quad z \in \mathcal{Z} \tag{23}$$

where the constraints are equivalent to those in Eq. (16). Intuitively, we want to minimize the net amount of expected improvement which corresponds to being very certain that we have found the minimum value in the environment.

This is a non-convex objective and is not easily handled by current optimizers. However, our approximate sequential path optimization approach can handle this objective by computing $r_j = \mathbb{E}[I_p]_j$ for each of the m prediction vertices in the environment with the current values of $\hat{x}$ and Σ obtained from the current path p_t. Note that we want to visit locations in the environment with high expected improvement to minimize the net expected improvement left in the environment so there is no negative sign in the expression for r_j
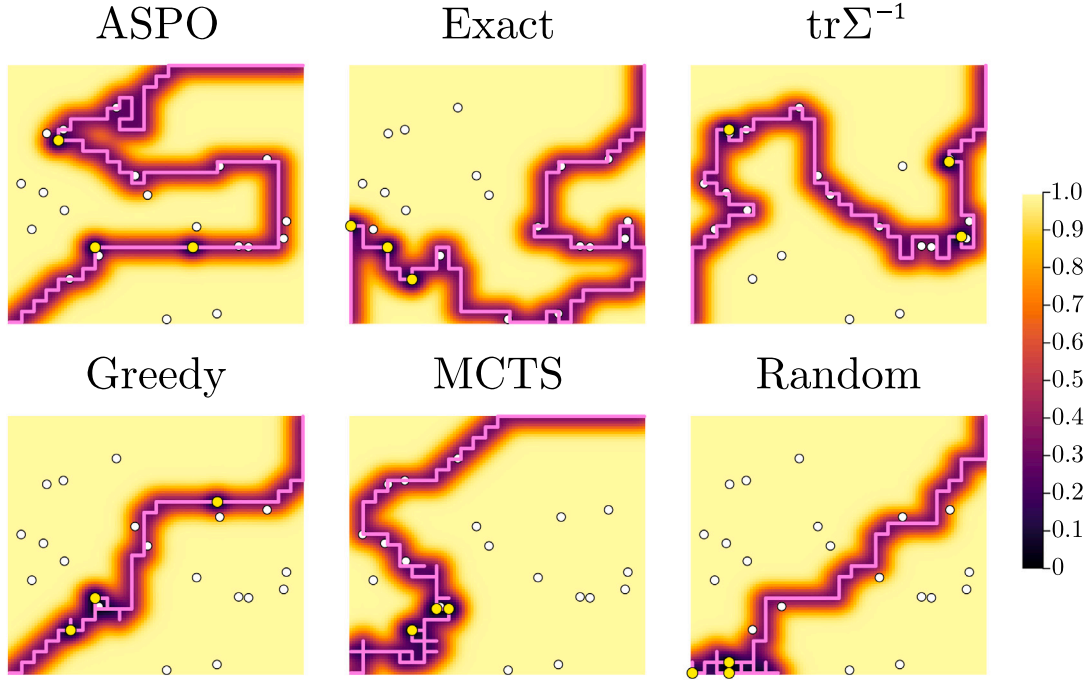
Fig. 4. Examples of trajectories produced by each of the six methods considered. For demonstration purposes, we have also included the multimodal sensing aspect discussed in Section 5.3. The magenta line indicates the trajectory of the agent through a graph of size $n = 625$. The $m = 20$ prediction locations are shown in white. The yellow points indicate where more accurate sensing locations were chosen. For visualization purposes, the variance is displayed across the entire grid rather than at the prediction locations only. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.3. Multimodal sensing

Up to this point, we have considered that each measurement taken by the agent is characterized by the same noise distribution $\mathcal{N}(0, \sigma^2)$. We can also consider a further extension of the problem to multimodal sensing domains where each measurement location also involves choosing a sensor type. The sensor type directly influences the variance σ_i^2 at the measured location. Adaptive informative path planning with multimodal sensing (AIPPMS) considers the case where the agent can choose between multiple sensing modalities (i.e. multiple sensors are available to choose from) [22,23]. This extension adds additional complexity by requiring that the agent reasons about the cost-benefit trade-off associated with the different sensing modalities.

To solve the AIPPMS problem, we first solve the AIPP problem with our approximate sequential path optimization approach and the respective objective presented in Section 4. We then use the resulting path as the starting point for the multimodal sensor selection problem. The multimodal sensor selection problem is a variant of the sensor selection problem presented by Joshi and Boyd. In this case, we assume we have k high-quality sensors to choose from with corresponding σ_i values ranging from $\sigma_{\min}$ to $\sigma_{\max}$. We constrain the multimodal sensor selection problem to only consider sensor locations along the path p. The multimodal sensor selection problem can then be stated as:

$$
\begin{aligned}
\text{minimize} \quad & \phi(s) \\
\text{subject to} \quad & \mathbf{1}^T s = k \\
& 0 \leq s_i \leq 1 \\
& s_i = 0 \quad \forall i \notin p
\end{aligned}
\tag{24}
$$

where s_i is the relaxed decision variable corresponding to the importance of placing high-value sensors at location i with

$$
\Sigma_s^{-1} = \left(\sigma_{\min}^{-2} \sum_{i=1}^{n} s_i a_i a_i^T + \Sigma_p^{-1} \right)
\tag{25}
$$

and

$$
\Sigma_p^{-1} = \left(\sigma_{\max}^{-2} \sum_{i=1}^{n} \left(\sum_{j \in N_{out}(i)} \hat{z}_{ij} \right) a_i a_i^T + \Sigma_x^{-1} \right)
\tag{26}
$$

and $\phi(s)$ is the corresponding IPP objective from Section 2.3 using the decision variable s. Note that Σ_p is the posterior covariance matrix from our path which has now become our prior for maximum a posteriori sensor selection. Additionally, notice that the individual values of σ_i are not used. We only require the bounds on their values. Similar to the procedure presented by Joshi and Boyd, once we solve this relaxed problem we then take the largest k values of s_i and assign them sensor values starting at $\sigma_{\min}$ corresponding to the largest s_i and proceeding onwards to $\sigma_{\max}$. In this way, we can think of each s_i as corresponding to the importance of that particular location. Locations with greater importance should be allocated greater resources. In the case where there are only two sensors to choose from and one needs to assign h measurement locations to the more accurate sensor $\sigma_{\min}$, we can simply set the largest h values of s_i to $\sigma_{\min}$. There are other versions of the multimodal sensor selection problem that can be formulated in a similar fashion.

5.4. Multi-agent IPP

Consider a set of M agents $\mathcal{A} = \{A_1, A_2, \ldots, A_M\}$. Each agent A_i is tasked with traversing the graph to collect measurements. The agents share their measurements, forming a collective measurement vector. For agent A_j, its path is $p^j = \{i_1^j, i_2^j, \ldots, i_{L_j}^j\}$, where L_j is the length of agent A_j's path. Along each path p^j, agent A_j makes measurements. The measurement at node i_k^j by agent A_j is represented as

$$
y_k^j = a_{i_k^j}^T x + v_{i_k^j}
\tag{27}
$$

which we can use to construct a unified measurement vector Y that combines measurements from all agents

$$
Y = \{y_1^1, \ldots, y_{L_1}^1, y_1^2, \ldots, y_{L_2}^2, \ldots, y_1^M, \ldots, y_{L_M}^M\}.
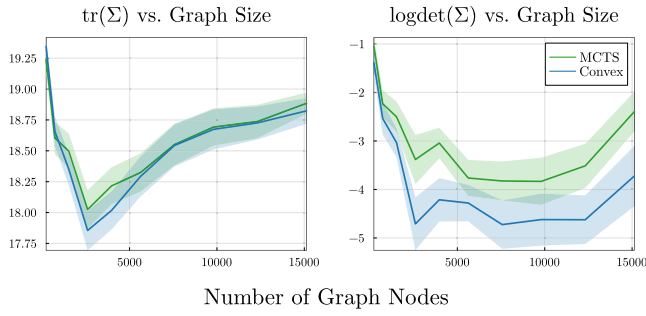\tag{28}
$$

Fig. 5. Comparison of the convex multimodal sensor selection against MCTS multimodal sensor selection. These comparisons were done with $k = 3$ high-quality sensors. We can see that the convex multimodal sensor selection tends to select more informative distributions of the high-quality sensors for both the A and D-IPP objectives.

Fig. 6. Results showing the average optimality gap and standard error as the budget is increased for the ASPO method with $N = 1600$. Here the gap is reported normalized by the lower bound for A-IPP and exponentiated for D-IPP.

The estimation of x is now based on the collective measurements from all agents similar to the sequential allocation method proposed by Singh et al. for multi-agent exploration [10]. The maximum a posteriori estimate from all agents measurements is:

$$\hat{x} = \bar{x} + \left(\sigma^{-2} \sum_{j=1}^{M} \sum_{k=1}^{L_j} a_{i_k^j} a_{i_k^j}^T + \Sigma_x^{-1} \right)^{-1} \sigma^{-2} \sum_{j=1}^{M} \sum_{k=1}^{L_j} (y_k^j - \bar{y}_k^j) a_{i_k^j} \qquad (29)$$

with covariance

$$\Sigma = \left(\sigma^{-2} \sum_{j=1}^{M} \sum_{k=1}^{L_j} a_{i_k^j} a_{i_k^j}^T + \Sigma_x^{-1} \right)^{-1} \qquad (30)$$

where $\bar{y}_k^j = a_{i_k^j}^T \bar{x}$.

Similarly, we now have for each agent A_j, a Boolean matrix $z^j \in \mathbf{R}^{n \times n}$ to parameterize its path p^j. Each $z_{ik}^j = 1$ if the edge from node i to node k is part of agent A_j's path, and 0 otherwise. Σ now takes into account the paths of all agents:

$$\Sigma = \left(\sigma^{-2} \sum_{j=1}^{M} \sum_{i=1}^{n} \left(\sum_{k \in N_{out}(i)} z_{ik}^j \right) a_i a_i^T + \Sigma_x^{-1} \right)^{-1}. \qquad (31)$$

The convex mixed integer problem for the multi-agent case is then:

$$\text{minimize} \quad \sum_{j=1}^{M} \phi(z^j)$$

$$\text{subject to} \quad z^j \in \mathcal{Z} \quad \forall j = 1, \ldots, M \qquad (32)$$

which can be handled by our approximate sequential path optimization in a similar fashion as presented in Section 4.

6. Results

We provide empirical evidence of the advantages of our approach over a variety of other baselines and methods from the literature. Namely, we directly compare our method with an implementation of the mixed integer formulation presented by Dutta et al. as well as the belief MDP formulation of Ott et al. [2,22]. Our experiments focus on the interplay between the number of nodes in the graph, the allotted budget, and the allowed runtime to evaluate the influence on the information objective of interest. All of the results presented in this section were run on a computer with an M1 Max and 64 GB of RAM.

6.1. Experimental setup

We consider environments modeled as grid-based graphs, but our approach scales to graphs with arbitrary structures. The graph is a $\sqrt{n} \times \sqrt{n}$ grid contained in an environment with constant edge lengths.
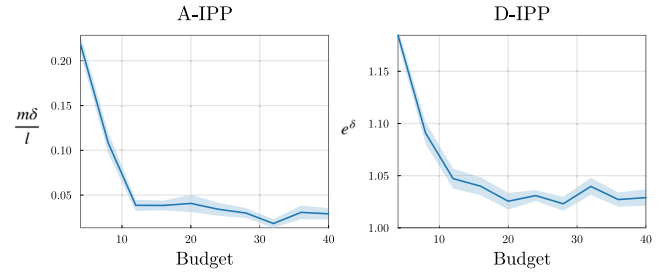
Our experimental setup is similar to that presented by Dutta et al.; to allow for consistent comparisons we use a prediction set fixed at $m = 20$ [2]. In all experiments, the start and end nodes are on diagonally opposite ends of the grid. The covariance structure of the environment is modeled using the squared exponential kernel

$$k(x, x') = \exp\left(-\frac{(x - x')^2}{2\ell^2} \right) \qquad (33)$$

where $\ell = 1.0$ is the length scale. Using this covariance structure, the prior density for x is given by $\mathcal{N}(0, \Sigma_x)$ where

$$\Sigma_x = \begin{bmatrix} k(x_1, x_1) & \ldots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \ldots & k(x_n, x_n) \end{bmatrix}. \qquad (34)$$

In all of the following experiments, we limit the maximum runtime for each method to 120 seconds to ensure even comparison across all approaches considered.

6.2. B-IPP as a surrogate objective for A-IPP

We first begin with the A-IPP problem and compare the exact mixed integer convex problem (MICP) presented in Eq. (16) with the exact MICP using the B-IPP $\mathbf{tr}(\Sigma^{-1})$ objective along with the mixed integer program (MIP) formulation from Dutta et al. [2]. For the exact MICP using the B-IPP objective we are optimizing the path with respect to the B-IPP objective and then evaluating the path with respect to the A-IPP objective. Note that the MIP formulation presented by Dutta et al. only considers the A-IPP problem and does not provide an equivalent formulation for the D-IPP objective. We first compare our results on small graphs ranging from $n = 4$ to $n = 3844$. Our results show that on small graph sizes ($n < 500$), all three of the methods perform similarly in both runtime and solution quality as shown in Fig. 1.

However, when we scale up to larger graph sizes, we can see that the $\mathbf{tr}(\Sigma^{-1})$ objective can scale much more efficiently. Note that all methods are restricted to the same 120 second time constraint and the best solution found is then returned. The MIP approach performs well for $n < 500$ but fails to scale to larger graphs. Most notably, the B-IPP objective proves to be an effective surrogate for the A-IPP problem. The reason for the improved performance is twofold. First, the $\mathbf{tr}(\Sigma^{-1})$ objective is an affine function of the decision variable z resulting in a mixed integer linear program. Second, by maximizing $\mathbf{tr}(\Sigma^{-1})$, we are trying to increase the magnitude of the eigenvalues of Σ_z^{-1}. We know that the eigenvalues of Σ_z^{-1} are the reciprocals of the eigenvalues of Σ_z (assuming Σ_z is invertible). Therefore, increasing the eigenvalues of Σ_z^{-1} will decrease the corresponding eigenvalues of Σ_z. While maximizing $\mathbf{tr}(\Sigma_z^{-1})$ indirectly influences the eigenvalues of Σ_z, the trace of Σ_z and the trace of Σ_z^{-1} are not simply reciprocals of each other. That is, the A and B-IPP objectives are not equivalent, but they are closely related and a meaningful performance improvement can be achieved using the B-IPP objective as a surrogate on larger problem instances. The goal of these experiments was to isolate the
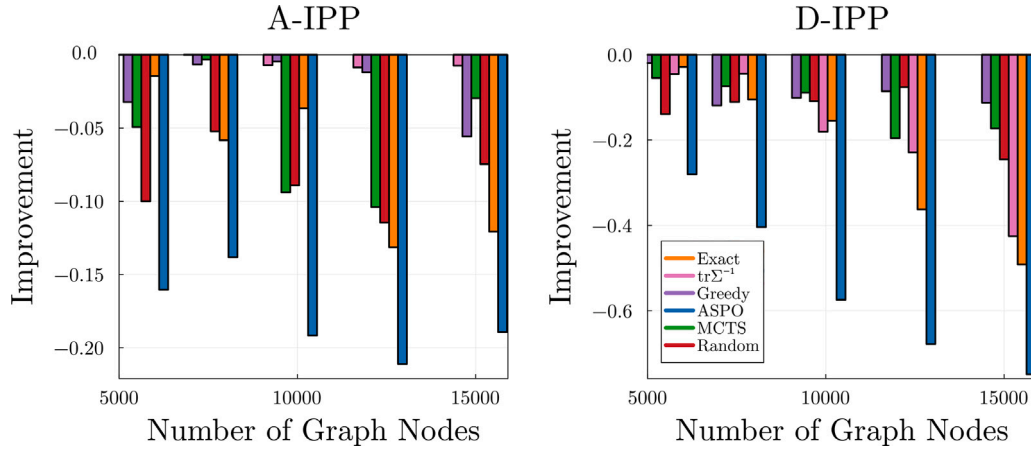
Fig. 7. Results from the local optimization procedure. The average improvement given by the reduction in objective value over 25 simulation runs is shown.

comparison between these three approaches to the A-IPP objective. We now proceed to scale our experiments to even larger graphs and include both A and D-IPP objectives with additional baselines to compare against.

6.3. A and D-IPP results

We compare the following approaches on both the A and D-IPP objectives:

1. **Exact**: the solution obtained from solving the exact MICP presented in Eq. (16).
2. **$\mathbf{tr}(\Sigma^{-1})$**: the solution obtained from solving the exact MICP using the B-IPP objective.
3. **MCTS**: the belief MDP approach presented by Ott et al. [22].
4. **Random**: starting from the initial vertex, the agent chooses randomly from the neighboring edges to then transition to the next vertex. The action space is constrained so that the agent can always reach the goal location within the specified budget constraint.
5. **Greedy**: starting from the initial vertex, the agent evaluates the improvement in the objective from taking each of the neighboring edges. The agent then selects the edge that results in the greatest performance and repeats this process. The action space is constrained so that the agent must always reach the goal location within the specified budget constraint.
6. **ASPO**: the approximate sequential path optimization approach from Section 4.

Fig. 2 shows the runtime and objective value as a function of graph size for the A-IPP objective evaluated on each of the six methods discussed above. Note that we used a modified version of MCTS where the number of iterations performed in the tree search was scaled proportional to the runtime budget used so far. The reason for doing this was to allow for better performance of MCTS by allowing for a dynamic adjustment of the iterations so that the entire 120 second runtime constraint would be used. As a result, we see that MCTS uses the full runtime constraint even for relatively small graphs. MCTS performs competitively with graphs of up to $n = 1000$ at which point the performance begins to degrade as the runtime constraint leads to fewer iterations since more individual action steps are required to make it to the goal location. We see that the $\mathbf{tr}(\Sigma^{-1})$ objective is the best performer on graph sizes of $n < 5000$ and then is outperformed by our ASPO method. ASPO performs similarly to the $\mathbf{tr}(\Sigma^{-1})$ objective on smaller graphs and outperforms all of the methods considered on larger graphs while also reducing the runtime by more than a factor of two relative to the $\mathbf{tr}(\Sigma^{-1})$ objective.

Fig. 3 shows the runtime and objective value as a function of graph size for the D-IPP objective evaluated on each of the six methods. Again, the $\mathbf{tr}(\Sigma^{-1})$ objective proves to perform quite well even when evaluated on the D-IPP objective. In this case, the exact and $\mathbf{tr}(\Sigma^{-1})$ methods perform similarly. MCTS exhibits similar performance to the A-IPP case where it performs competitively on small graph sizes and then degrades on larger graph sizes due to the runtime constraints. ASPO outperforms both the exact and $\mathbf{tr}(\Sigma^{-1})$ methods while again reducing runtime by more than a factor of two and scaling performance to the largest graph sizes considered.

Examples of the trajectories produced by each of the methods are shown in Fig. 4. In these example trajectories, we have also included the multimodal sensing aspect of the problem introduced in Section 5.3. MCTS used the method described by Ott et al. to select locations for the high-quality sensors [22]. The random method selected high-quality sensor locations from its action space and was limited to placing $k = 3$ high-quality sensors. The other methods all used the convex formulation provided in Section 5.3. From the trajectories, we can see that the random solution is the worst performer and stays near the starting location (bottom left corner) until it reaches the budget limit and must head to the goal location leaving much of the environment unexplored. The greedy method slightly improves upon the random method, but is unable to coordinate its actions several steps into the future therefore leading to inefficient exploration. MCTS improves upon the greedy solution; however, much of its exploration is limited to small areas around where it has already visited. The exact and $\mathbf{tr}(\Sigma^{-1})$ methods are able to better distribute the measurement locations resulting in more efficient exploration. Lastly, the approximate sequential path optimization method tries to fully cover the environment by distributing its measurements accordingly and outperforms the exact and $\mathbf{tr}(\Sigma^{-1})$ methods in terms of the best (lowest) value of the A-IPP objective.

6.4. Quality of convex sensor selection vs. MCTS

Fig. 5 compares the multimodal sensor selection methods of Ott et al. and the method we present in Section 5.3. For these experiments, we started with the paths that were produced from the MCTS policy. To construct these paths, the MCTS policy chooses from an action space that includes both movement and sensing actions. Therefore, each point along the trajectory includes a location and sensor type to be used. To compare the sensor selection methods, we left the MCTS path solution constant and used this as the initial path for the convex multimodal sensor selection problem. We then evaluated both sensor selection methods on the A and D-IPP objectives (i.e. the paths were fixed for both methods, but the sensor types were changed). Fig. 5 shows that the convex multimodal sensor selection method we propose outperforms that of MCTS. Note that these comparisons were done with $k = 3$ high-quality sensors.
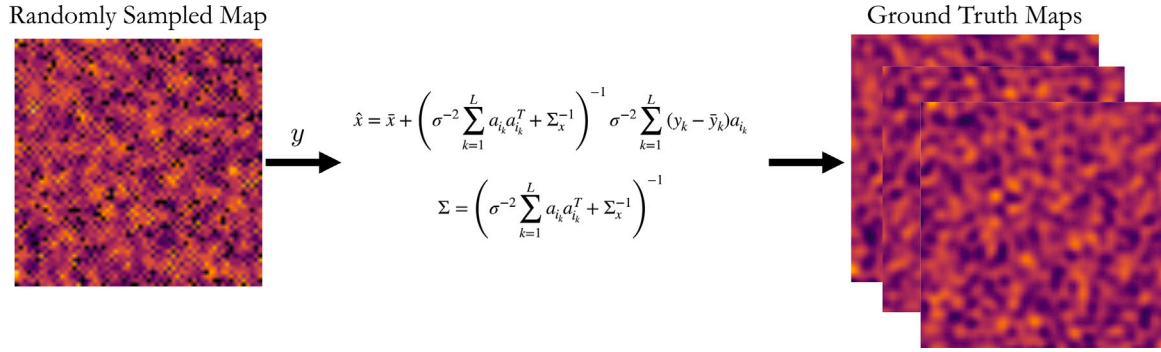
Randomly Sampled Map

Ground Truth Maps

$$\hat{x} = \bar{x} + \left(\sigma^{-2}\sum_{k=1}^{L} a_{i_k}a_{i_k}^T + \Sigma_x^{-1}\right)^{-1}\sigma^{-2}\sum_{k=1}^{L}(y_k - \bar{y}_k)a_{i_k}$$

$$\Sigma = \left(\sigma^{-2}\sum_{k=1}^{L} a_{i_k}a_{i_k}^T + \Sigma_x^{-1}\right)^{-1}$$

Fig. 8. Illustration showing the construction of the ground truth maps used in our adaptive IPP experiments. Note the length scale in Eq. (33) controls the amount of spatial correlation in the environment.
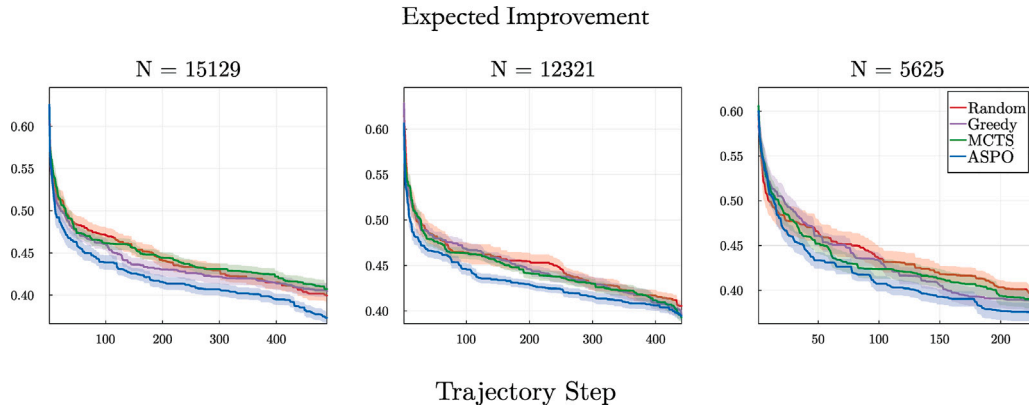

Fig. 9. Results showing the net expected improvement normalized by graph size as the trajectory progresses for each of the methods that support adaptive objectives.

6.5. Optimality gap

Fig. 6 shows the empirical results of the optimality gap using our approximate sequential path optimization method as the budget constraint is varied. We can see that as the budget is increased, our solution decreases the optimality gap. Here, $\frac{m\delta}{l}$ quantifies how close the solution is to the lower bound for the A-IPP objective. The factor of m is included for appropriate scaling with the definition of δ given in Eq. (20). For the D-IPP objective, we use e^{δ} which is the ratio of mean radii of the confidence ellipsoid [1]. Additionally, we can see that our proposed method is always within at least 25% of the lower bound. In reality, we are often much closer to the true optimal value since the lower bound is not necessarily the true optimal value.

6.6. Local optimization

As discussed by Joshi and Boyd and Diamond et al., we can perform a local optimization step to greedily improve our approximate solution [1,29]. This local optimization procedure is the polish step in the relax-round-polish method presented by Diamond et al. [29]. Starting from the approximate solution $\hat{z}$ presented in Section 4, we check other paths that can be derived from $\hat{z}$ by swapping one of the k nodes along the path with one of the one-hop neighbors available. In this case, we define a valid one-hop neighbor of node i to be a node j such that $j \in N_{out}(i)$ and if node i were removed from the path and node j were added then the total length of the path would not change. Equivalently, if $l_A = \sum_{i=1}^{n}\sum_{j \in N_{out}(i)}\hat{z}_{ij}$ represents the total length of path A before a valid swap occurs then l_A must equal l_B where path B contains a valid swap derived from path A.

Given the current approximate solution $\hat{z}$, we randomly sample a node j from the path and check to see if any of its valid one-hop neighbors improve the objective when swapped with node j. If so, we

swap the nodes and continue the random search. In theory, we could search over all possible one-hop neighbors of the path and continue this search until no swap is found to improve the objective. The resulting solution would be the $2 - opt$ solution because exchanging any node on the path with any valid one-hop neighbor will not improve the solution. Doing so can involve a large number of local optimization steps, so in practice, we can simply limit the number of steps taken to N_{loc}.

The results from this local optimization polishing procedure are shown in Fig. 7. We can see that the local optimization tends to improve the performance of each of the methods. In some cases, we see that the local optimization has a greater effect than others. Namely, it has a consistently larger effect on the exact and ASPO methods for both the A and D-IPP objectives. The $\mathbf{tr}(\Sigma^{-1})$ method benefits from an improvement on the D-IPP objective, while only receiving a small improvement on the A-IPP objective. Typically, methods that start with reasonable solutions to begin with, such as ASPO and the exact method, can achieve a significant performance boost by using the simple local optimization procedure since only valid one-hop swaps are considered. However, this is not always the case as indicated by the difference in improvement for the $\mathbf{tr}(\Sigma^{-1})$ method between A and D-IPP objectives.

6.7. Other objectives

As discussed in Section 5.2, our ASPO method can handle the adaptive version of the IPP problem where the objective depends on the received sample values. The previous experiments were not concerned with the actual sensor observation values received at each of the nodes in the environment, but in the adaptive IPP problem, these sensor observations now become part of the adaptive objective.

For these experiments, we construct a ground truth spatial map of the environment containing the true quantity of interest. The environment is modeled as an $\sqrt{n} \times \sqrt{n}$ grid where each cell contains a value
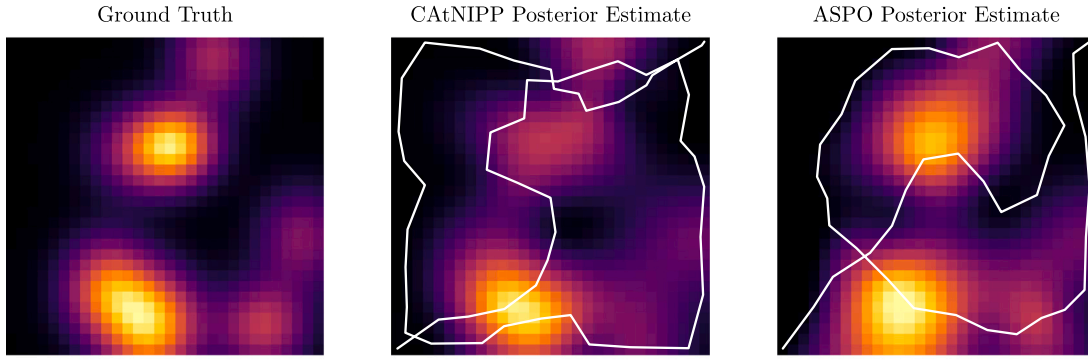
Ground Truth	CAtNIPP Posterior Estimate	ASPO Posterior Estimate

Fig. 10. Examples of trajectories planned by the CAtNIPP trajectory sampling variant and the ASPO method on a graph with 400 nodes and available budget of 6.

Table 1

Comparison between CAtNIPP Greedy (CG), CAtNIPP Trajectory Sampling (CTS), and ASPO IPP solvers on the AIPP problem using the environment setup presented by Cao et al. with 400 nodes, 20 edge connections, and 4 trajectory samples for CTS [30]. The average covariance matrix trace $\mathbf{tr}(\Sigma)$ in high interest areas is reported after running out of budget (standard error in parentheses). T(s) is the average total planning time in seconds. Results from 100 simulation runs for each budget.

Method	Budget 4		Budget 6		Budget 8		Budget 10		Budget 12	
	$\mathbf{tr}(\Sigma)$	T(s)	$\mathbf{tr}(\Sigma)$	T(s)	$\mathbf{tr}(\Sigma)$	T(s)	$\mathbf{tr}(\Sigma)$	T(s)	$\mathbf{tr}(\Sigma)$	T(s)
CG	19.67($\pm$1.57)	**0.15**	5.42($\pm$0.29)	**0.19**	2.57($\pm$0.14)	**0.26**	1.58($\pm$0.09)	**0.33**	1.24($\pm$0.07)	**0.42**
CTS	15.94($\pm$1.18)	44.04	4.94($\pm$0.23)	62.35	2.27($\pm$0.12)	88.78	1.54($\pm$0.08)	111.88	1.24($\pm$0.07)	143.94
ASPO	**14.02($\pm$0.87)**	0.69	**4.79($\pm$0.28)**	1.26	**2.24($\pm$0.13)**	1.96	**1.28($\pm$0.11)**	2.61	**0.98($\pm$0.11)**	3.33

for the true quantity of interest. To construct a spatially-correlated environment, we first start by randomly generating a value for each grid cell. We then use these randomly generated values as the y values in Eq. (3) using the covariance structure from Eq. (33). This gives us the mean and covariance of the Gaussian process which we use to sample different ground truth maps for different simulation runs. An illustration of this process is shown in Fig. 8.

Fig. 9 shows our experiment results for the adaptive IPP problem. For this set of experiments, we only compared the Random, Greedy, MCTS, and ASPO methods since these were the only methods that supported adaptive objectives. As we can see, ASPO outperforms the other methods in all graph sizes considered and consistently results in a net expected improvement remaining in the environment of 0.4 or less when normalized by graph size. As expected, we see the random method tends to perform the worst. In large graphs, we see that MCTS and Greedy only perform marginally better due to the significant increase in the size of the search space.

These results empirically support the teleportation approximation used by the ASPO method to estimate the value of visiting different regions of the environment. In addition, it demonstrates the flexibility of the ASPO method to support both adaptive and non-adaptive objectives, as well as incorporating multimodal sensing. In this way, the ASPO method can outperform the other methods considered on the IPP, AIPP, and AIPPMS problems considered.

6.8. Comparison with attention-based network approach

We also compared our approximate method with the Context-Aware Attention-based Network for Informative Path Planning (CAtNIPP) approach presented by Cao et al. [30]. CAtNIPP is a state-of-the-art deep reinforcement learning based approach using attention-based neural networks to learn policies that balance short-term exploitation and long-term exploration for the AIPP problem.

For even comparison between our two methods, we used the graph and environment setup described by Cao et al. [30]. Specifically, a Matérn $3/2$ Kernel with length scale of 0.45. The ground truth environment is constructed by averaging 8 to 12 random 2-dimensional Gaussian distributions in the unit square $[0, 1]^2$, to construct the true interest map [30].

The objective is to reduce the variance in high-interest areas defined as regions that have true quantities of interest greater than or equal to 0.4. To effectively solve this variant of the AIPP problem, solution methods must balance the tradeoff between collecting additional measurements in high-interest areas they have previously discovered with exploring new areas of the environment they have not yet visited. Cao et al. present two different variants of CAtNIPP. The greedy variant works in the standard reinforcement learning manner where the agent selects the action with the highest activation in its policy. The trajectory sampling variants plan a 15-step trajectory and execute the first 3 steps before replanning.

Table 1 presents the results across several different budgets. We see that across all of the budgets, our ASPO method consistently achieves the lowest variance in high-interest areas showing that it more effectively balances the exploration exploitation tradeoff. We see that the CAtNIPP Greedy variant has the lowest planning time across all of the methods considered but also is consistently the worst performer. ASPO is quite close to the Greedy variant in terms of planning time, while the trajectory sampling variant requires significantly more planning time. Fig. 10 also shows examples of trajectories planned by the CAtNIPP trajectory sampling variant and ASPO in addition to the posterior estimates after executing the trajectories.

It is important to note that this comparison was performed on the specific environment that CAtNIPP was trained on. CAtNIPP is sensitive to changes in the hyperparameters of the Gaussian process kernel function and the upper bound confidence that defines the high-interest areas. Therefore, CAtNIPP would require retraining if any of these parameters were changed significantly. Cao et al. report that it took around 24 hours training on four GPUs to converge. Therefore, one of the key strengths of ASPO over neural network based approaches is that no retraining is required when any of the Gaussian process hyperparameters, graph structures, budgets, objectives, or confidence intervals are changed.

6.9. Multi-agent example

Fig. 11 illustrates the application of the ASPO method in a multi-agent Informative Path Planning (IPP) scenario, where $M = 3$ agents are deployed in an environment that includes obstacles. Each agent A_i is assigned the task of traversing through the environment to collect
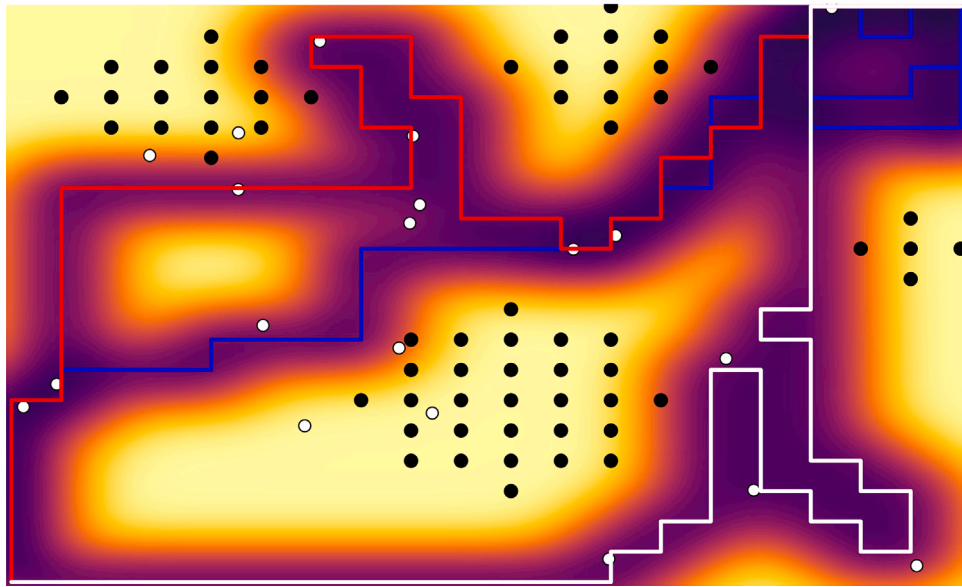
Multi-agent IPP

Fig. 11. Example of the multi-agent IPP solution produced with our ASPO method. Obstacles in the environment are shown by black circles. The $m = 20$ prediction locations are shown by white circles. The three agent trajectories are shown in red, blue, and white. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

measurements. The measurements gathered by the individual agents are then shared among them, resulting in a collective measurement vector that represents the combined data from all agents.

We can see that the agents are able to distribute their exploration throughout the environment to effectively split up the environment into thirds leading to more efficient exploration. Notably, the solution to this multi-agent problem, computed using our ASPO method, was achieved in less than 5 seconds, on a graph with 400 nodes. This highlights the efficiency and scalability of the ASPO method, particularly in complex environments with multiple agents and obstacles.

7. Conclusion

This study focused on planning informative paths in graphs using information-theoretic objectives. We demonstrated the challenge that exact solution methods such as mixed integer programs have in scaling to larger graph sizes. We introduced the B-IPP objective and showed that its formulation as a mixed integer linear program positions it as an effective surrogate objective for both the A and D-IPP problems. We proposed a novel approximate sequential path optimization method using dynamic programming to produce approximate solutions to the IPP problem. Our ASPO method achieves a balance between computational feasibility and solution quality. By leveraging the strengths of dynamic programming, our method iteratively constructs paths, incorporating the updated information distribution at each step.

We provided a bound on the optimality of our approximate method, confirming that our approximations are not only efficient but also consistently near optimal. Furthermore, the flexibility of our method is evident in its capability to handle adaptive objectives, multimodal sensing, and multi-agent extensions of the classic IPP problem. This flexibility highlights the capability of our method in the identification and prioritization of pertinent regions based on received measurement values. We demonstrated the efficacy of our ASPO method in larger graphs showing that solutions could be produced on graphs with more than 4000 nodes and 15,000 edges within seconds. The flexibility and efficiency of our method are crucial for real-world settings in dynamic environments where the relevance of information can change rapidly.

7.1. Limitations and future work

While our method has proven effective, there are some key limitations that present opportunities for future research. First, our approach assumes that the location of the agent is fully observable, the current energy state of the agent is known, and the amount of budget used by specific actions is deterministic. Incorporating uncertainty in these aspects would make our method more robust and better aligned with real-world scenarios where localization errors, energy consumption variability, and unexpected environmental changes are common. Additionally, our current work assumes full knowledge of these variables, and introducing uncertainty in these areas could lead to more sophisticated and resilient planning strategies.

Another promising area for future research involves exploring more complex multi-agent scenarios. Specifically, we aim to investigate cases where communication between agents is limited or unreliable, which is a common challenge in real-world applications. Additionally, introducing adversarial agents into the problem space, which may attempt to disrupt or compete with the planning agents, could add another layer of complexity and realism to the problem. Addressing these challenges will likely require the development of new approximate methods that can maintain scalability while handling the increased combinatorial complexity introduced by these factors.

CRediT authorship contribution statement

Joshua Ott: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Mykel J. Kochenderfer:** Writing – review & editing, Supervision, Resources, Project administration, Conceptualization. **Stephen Boyd:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

https://github.com/sisl/InformativePathPlanning.

Acknowledgments

We would like to thank Shamak Dutta for his insightful discussions and support in comparing with his previous work. We would also like to thank Lauren Ho-Tseung for her contributions and support in developing this method.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.robot.2024.104814.

References

[1] S. Joshi, S. Boyd, Sensor selection via convex optimization, IEEE Trans. Signal Process. 57 (2) (2008) 451–462.

[2] S. Dutta, N. Wilde, S.L. Smith, Informative path planning in random fields via mixed integer programming, in: IEEE Conference on Decision and Control, CDC, IEEE, 2022, pp. 7222–7228.

[3] A. Meliou, A. Krause, C. Guestrin, J.M. Hellerstein, Nonmyopic informative path planning in spatio-temporal models, in: AAAI Conference on Artificial Intelligence, Vol. 10, No. 4, AAAI, 2007, pp. 16–7.

[4] P. Morere, R. Marchant, F. Ramos, Sequential Bayesian optimization as a POMDP for environment monitoring with UAVs, in: IEEE International Conference on Robotics and Automation, ICRA, 2017, pp. 6381–6388.

[5] M. Popović, T. Vidal-Calleja, G. Hitz, J.J. Chung, I. Sa, R. Siegwart, J. Nieto, An informative path planning framework for UAV-based terrain monitoring, Auton. Robots 44 (6) (2020) 889–911.

[6] A. Vashisth, J. Ruckin, F. Magistri, C. Stachniss, M. Popovic, Deep reinforcement learning with dynamic graphs for adaptive informative path planning, IEEE Robot. Autom. Lett. (2024).

[7] T. Yu, B. Deng, J. Gui, X. Zhu, W. Yao, Efficient informative path planning via normalized utility in unknown environments exploration, Sensors 22 (21) (2022) 8429.

[8] K. Jakkala, S. Akella, Multi-robot informative path planning from regression with sparse Gaussian processes, 2023, arXiv preprint arXiv:2309.07050.

[9] A. Krause, A. Singh, C. Guestrin, Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies, J. Mach. Learn. Res. 9 (2) (2008).

[10] A. Singh, A. Krause, C. Guestrin, W.J. Kaiser, Efficient informative sensing using multiple robots, J. Artificial Intelligence Res. 34 (2009) 707–755.

[11] D. Strawser, B. Williams, Motion planning under uncertainty with complex agents and environments via hybrid search, J. Artificial Intelligence Res. 75 (2022) 1–81.

[12] C. Bäckström, P. Jonsson, S. Ordyniak, Cost-optimal planning, delete relaxation, approximability, and heuristics, J. Artificial Intelligence Res. 70 (2021) 169–204.

[13] P. Boström-Rost, D. Axehill, G. Hendeby, On global optimization for informative path planning, IEEE Control Syst. Lett. 2 (4) (2018) 833–838.

[14] S. Song, A. Rodriguez, M. Teodorescu, Trajectory planning for autonomous nonholonomic vehicles for optimal monitoring of spatial phenomena, in: International Conference on Unmanned Aircraft Systems, IEEE, 2015, pp. 40–49.

[15] G. Hitz, E. Galceran, M.-È. Garneau, F. Pomerleau, R. Siegwart, Adaptive continuous-space informative path planning for online environmental monitoring, J. Field Robotics 34 (8) (2017) 1427–1449.

[16] G. Francis, L. Ott, R. Marchant, F. Ramos, Occupancy map building through Bayesian exploration, Int. J. Robot. Res. 38 (7) (2019) 769–792.

[17] A. Asgharivaskasi, S. Koga, N. Atanasov, Active mapping via gradient ascent optimization of Shannon mutual information over continuous SE (3) trajectories, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2022, pp. 12994–13001.

[18] R. Marchant, F. Ramos, S. Sanner, et al., Sequential Bayesian optimisation for spatial-temporal monitoring, in: Conference on Uncertainty in Artificial Intelligence, UAI, 2014, pp. 553–562.

[19] P. Morere, R. Marchant, F. Ramos, Continuous state-action-observation POMDPs for trajectory planning with Bayesian optimisation, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2018, pp. 8779–8786.

[20] I.M.R. Fernández, C.E. Denniston, D.A. Caron, G.S. Sukhatme, Informative path planning to estimate quantiles for environmental analysis, IEEE Robot. Autom. Lett. (2022).

[21] J. Rückin, L. Jin, M. Popović, Adaptive informative path planning using deep reinforcement learning for UAV-based active sensing, in: IEEE International Conference on Robotics and Automation, ICRA, 2022.

[22] J. Ott, E. Balaban, M.J. Kochenderfer, Sequential Bayesian optimization for adaptive informative path planning with multimodal sensing, in: IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2023, pp. 7894–7901.

[23] S. Choudhury, N. Gruver, M.J. Kochenderfer, Adaptive informative path planning with multimodal sensing, in: International Conference on Automated Planning and Scheduling, Vol. 30, ICAPS, 2020, pp. 57–65.

[24] M.J. Kochenderfer, T.A. Wheeler, Algorithms for Optimization, MIT Press, 2019.

[25] C.E. Miller, A.W. Tucker, R.A. Zemlin, Integer programming formulation of traveling salesman problems, J. ACM 7 (4) (1960) 326–329.

[26] S.P. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[27] P. Vansteenwegen, W. Souffriau, D. Van Oudheusden, The orienteering problem: A survey, European J. Oper. Res. 209 (1) (2011) 1–10.

[28] A. Gunawan, H.C. Lau, P. Vansteenwegen, Orienteering problem: A survey of recent variants, solution approaches and applications, European J. Oper. Res. 255 (2) (2016) 315–332.

[29] S. Diamond, R. Takapoui, S. Boyd, A general system for heuristic minimization of convex functions over non-convex sets, Optim. Methods Softw. 33 (1) (2018) 165–193.

[30] Y. Cao, Y. Wang, A. Vashisth, H. Fan, G. Sartoretti, Context-aware attention-based network for informative path planning, in: 6th Annual Conference on Robot Learning, 2022.

Joshua Ott is a Ph.D. Candidate in Aeronautics & Astronautics at Stanford University. Joshua received his M.S. in Aeronautics & Astronautics from Stanford University in 2021 and his B.S. in Mechanical Engineering from the University of California, Berkeley in 2020. He is currently serving on Active Duty in the United States Air Force through the DAWN-ED Ph.D. fellowship. Joshua's current research in the Stanford Intelligent Systems Lab (SISL) focuses on decision making under uncertainty for autonomous exploration of unknown environments.

Mykel Kochenderfer is an Associate Professor of Aeronautics and Astronautics at Stanford University. He received a B.S. and an M.S. from Stanford University, and a Ph.D. from the University of Edinburgh. He was a technical staff member at the MIT Lincoln Laboratory. He is the director of the Stanford Intelligent Systems Laboratory (SISL), conducting research on advanced algorithms and analytical methods for designing robust decision making systems in applications such as air traffic control, unmanned aircraft, and automated driving. He authored the textbooks Decision Making under Uncertainty: Theory and Application, Algorithms for Optimization, and Algorithms for Decision Making.

Stephen Boyd is a Professor of Electrical Engineering at Stanford University. He received an AB degree in Mathematics from Harvard University, and a Ph.D. in Electrical Engineering and Computer Science from the University of California, Berkeley. He is the director of the Information Systems Laboratory (ISL), where his research focuses on convex optimization, control systems, and signal processing. He is the author of several textbooks, including Convex Optimization and Linear Matrix Inequalities in System and Control Theory.