An Extended Bayesian Optimization Approach to Decentralized Swarm Robotic Search

Swarm robotic search aims at searching targets using a large number of collaborating simple mobile robots, with applications to search and rescue and hazard localization. In this regard, decentralized swarm systems are touted for their coverage scalability, time efficiency, and fault tolerance. To guide the behavior of such swarm systems, two broad classes of approaches are available, namely, nature-inspired swarm heuristics and multi-robotic search methods. However, the ability to simultaneously achieve efficient scalability and provide fundamental insights into the exhibited behavior (as opposed to exhibiting a black-box behavior) remains an open problem. To address this problem, this paper extends the underlying search approach in batch-Bayesian optimization to perform search with embodied swarm agents operating in a (simulated) physical 2D arena. Key contributions lie in (1) designing an acquisition function that not only balances exploration and exploitation across the swarm but also allows modeling knowledge extraction over trajectories and (2) developing its distributed implementation to allow asynchronous task inference and path planning by the swarm robots. The resulting collective informative path planning approach is tested on target-search case studies of varying complexity, where the target produces a spatially varying (measurable) signal. Notably, superior performance, in terms of mission completion efficiency, is observed compared to exhaustive search and random walk baselines as well as a swarm optimization-based state-of-the-art method. Favorable scalability characteristics are also demonstrated. [DOI: 10.1115/1.4046587]

Keywords: swarm robotic search, informative path planning, Bayesian optimization, Gaussian process, artificial intelligence

1 Introduction

Swarm robotic search is concerned with searching for or localizing targets in unknown environments with a large number of collaborating robots. There exists a class of search problems in which the goal is to find the source or target with maximum strength (often in the presence of weaker sources) and where each source emits a spatially varying signal. Potential applications include source localization of gas leakage [1], nuclear meltdown tracking [2], chemical plume tracing [3], and magnetic field and radio source localization [4, 5]. In such applications, decentralized swarm robotic systems have been touted to provide mission efficiency, fault tolerance, and scalable coverage advantages [6–8] compared to sophisticated standalone systems. Decentralized search subject to a signal with unknown spatial distribution usually requires both task inference and planning, which must be undertaken in a manner that maximizes search efficiency and mitigates inter-robot conflicts. This in turn demands decision algorithms that are computationally lightweight (i.e., amenable to onboard execution) [9], preferably explainable [10], and scalable [11]—it is particularly challenging to meet these characteristics simultaneously.

In this paper, we perceive the swarm robotic search process to consist of creating/updating a model of the signal environment and deciding future waypoints thereof, so as to collectively find the target source (location with maximum signal strength) as fast as possible. Specifically, we design, implement, and test a novel decentralized algorithm founded on a Bayesian search formalism.

This algorithm tackles the exploration/exploitation balance over trajectories (as opposed to over points, which is typical in non-embodied search) while allowing asynchronous decision-making. In this context, we also explicitly consider other constraints attributed to the embodiment of the search process, e.g., individual robot’s speed and range constraints. The remainder of this section briefly surveys the literature on swarm search algorithms and converges on the contributions of this paper.

1.1 Swarm Robotic Search. In time-sensitive search applications under complex signal distributions, a team of robots can broaden the scope of operational capabilities through distributed remote sensing, scalability, and parallelism (in terms of task execution and information gathering) [12]. The multi-robot search paradigm [11] uses concepts such as cooperative control, model-driven strategies [13], Bayesian filter by incorporating mutual information [14], strategies based on local cues [15], and uncertainty reduction methods [16]. Scaling these methods from the multi-robotic (<10 agents [11]) to the swarm-robotic level (10–100 agents) often becomes challenging in terms of online computational tractability.

A different class of approaches that is dedicated to guiding the search behavior for larger teams is based on nature-inspired swarm intelligence (SI) principles [17–19]. SI-based heuristics have been used to design algorithms both for search in non-embodied n-dimensional space (e.g., particle swarm optimization) and for swarm robotic search [20, 21]. Majority of the latter methods are targeted at localizing a single source [9, 22]. A notable exception is the Glowworm optimization-based algorithm reported by Krishnanand et al. [18]. This approach was shown to handle multi-modal source localization by assuming robots are...
initially distributed in the search space, with its effectiveness relying on the usage of adaptive parameters (e.g., changing inertia weight) [22]. The localization of the maximum strength source in the presence of other weaker sources (i.e., given a multi-modal spatial signal-distribution), without making assumptions such as distributed starting points, remains a challenging problem.

Translating optimization processes: Similar in principle to some SI approaches, here we aim to translate an optimization strategy [23], namely, Bayesian optimization, to perform search in the physical 2D environment. In doing so, it is important to appreciate two critical differences between these processes: (1) movement cost: unlike optimization, in swarm robotic search, moving from one point to another may require a different energy/time cost depending upon the environment (distance, barriers, etc.) separating the current and next waypoints. (2) Sampling over paths: robots usually gather multiple samples (signal measurements) over the path from one waypoint to the next (as sampling frequency >> waypoint frequency), unlike in optimization where we sample only at their next planned point. This “sampling over paths” characteristic has received minimal attention in existing SI-based approaches.

Moreover, with SI-based methods, the resulting emergent behavior, although often competitive, raises questions of dependability (due to the use of heuristics) and mathematical explainability [24]. The search problem can be thought of as comprising two main steps: task inference (identifying/updating the signal spatial model) and task selection (waypoint planning). In SI methods, the two steps are not separable, and a spatial model is not explicit. In our proposed approach, the processes are inherently decoupled—robots exploit Gaussian processes (GPs) to model the signal distribution knowledge (task inference) and solve a 2D optimization over a special acquisition function to decide waypoints (task selection). Such an approach is expected to provide explainability, while preserving computational tractability.

1.2 Objective of This Paper. This paper is an extension of our recent work presented in the ASME 2019 IDETC/CIE conference [25]. In this paper, we develop (an explainable) decentralized and asynchronous swarm robotic search algorithm, subject to the following assumptions: (i) all robots are equipped with precise localization and (ii) each robot can communicate their knowledge, state, and decisions with all neighbors (full observability) at waypoints. In asynchronous decision-making, agents/robots take decision in an event-driven scheme, as opposed to a synchronous approach where all robots need to take decisions at fixed intervals. The asynchronous decision-making is critical in most real-world settings, due to the presence of stochastic action effects and imperfect and unreliable communication [26]. In addition, it has been shown that having asynchronous parallel sampling in Bayesian optimization (the motivating algorithm behind our proposed search method) can improve the optimization progress in comparison to synchronous implementations [27]. Decentralized decision-making here relates to how each swarm robotic agent independently plans its immediate future waypoint.

Within this context, the primary contributions of this paper lie in the following developments: (1) a novel decentralized algorithm (Bayes-Swarm) that extends Gaussian process modeling (to update over trajectories) and integrates physical robot constraints and other robots’ decisions to perform informative path planning—simultaneously mitigating knowledge uncertainty and getting closer to the source—and (2) a simulated parallelized implementation of Bayes-Swarm to allow asynchronous search planning over complex multi-modal signal distributions.

The remaining portion of the paper is organized as follows: the next section presents the problem definition and GP modeling. Then, our proposed decentralized algorithm (Bayes-Swarm) is described. Numerical experiments and results, encapsulating the performance of these methods on different-sized swarm and a parametric analysis of the proposed decentralized method, are then presented. The paper ends with concluding remarks.

2 Background

2.1 Gaussian Process Model. GP models [28] are probabilistic surrogate models that have been used successfully in different applications such as modeling the objective function in Bayesian optimization [29]. If we have a set of observations of an environment, \( d = y_i, i = 1, \ldots, n \), then we can write the following equation by assuming that the observed values \( y \) differ from the function \( f(x) \) values by an additive noise \( \epsilon \), where \( x \) denotes an input vector:

\[
y = f(x) + \epsilon
\]  

(1)

By assuming that the noise follows an independent, identically distributed Gaussian distribution with zero mean and variance \( \sigma^2_\epsilon \), the function \( f(x) \) can be estimated by a GP with mean function \( \mu(x) \) and covariance kernel \( \sigma^2(x) \) given by

\[
f(x) \sim GP(\mu(X), \sigma^2(X))
\]  

(2)

where

\[
\mu(x) = \Lambda(x)(y - \Phi \hat{\beta})
\]  

(3)

\[
\sigma^2(x) = k(x, x) - \Lambda(x) k_0(x)
\]  

(4)

\[
\Lambda(x) = k_0(x)^T [K + \sigma^2_\epsilon I]^{-1}
\]  

(5)

Here, \( \Phi \) is the vector of explicit basis functions and \( K = K(X, X) \) is the covariance function matrix such that \( (K_{ij} = k(x_i, x_j) \) and \( k_0(x) = k(x_1, x), \ldots, k(x_n, x) \)\). In this paper, the hyper-parameters of the GP model are optimized by maximizing the log-likelihood \( P \) as a function of \( \hat{\beta}, \theta, \sigma^2_\epsilon \):

\[
\hat{\beta}, \theta, \sigma^2_\epsilon = \text{arg max} \log P(y|X, \beta, \theta, \sigma^2_\epsilon)
\]  

(6)

where

\[
\log P(y|X, \beta, \theta, \sigma^2_\epsilon) = -\frac{1}{2} (y - \Phi \hat{\beta})^T \Lambda(x)^{-1} (y - \Phi \hat{\beta}) - \frac{1}{2} \log(\det(\Lambda(x)))
\]  

(7)

3 Swarm Bayesian Algorithm

3.1 Bayes-Swarm: Overview. The robot behaviors including its motion, communication, and decision-making are illustrated in Fig. 1 and the pseudocode of our proposed decentralized Bayes-Swarm algorithm is depicted in Algorithm 1. Each robot in a team of size \( N_r \) is assumed to run the Bayes-Swarm algorithm at each decision-making step (i.e., after reaching its waypoint) to take the best action by maximizing an acquisition function that guides the team to the source location over the course of the operation. Importantly, these decision-making instances need not be synchronized across robots, unlike several other existing decentralized implementations. Before elaborating on the mathematical formulation of each component of the Bayes-Swarm algorithm, we provide here a brief description of how the overall algorithm works, using Fig. 1 as reference. At the beginning of the mission, the robots do not have any observations from the environment, and thus no belief model to follow (as the default setting); prior knowledge, if available in a practical application, can however be readily incorporated as a prior belief model in our formulation. By default, in the “Select First Waypoint” block in Fig. 1, each robot chooses a waypoint such that the heading directions of the team are somewhat uniformly distributed over the domain of interest. Then, each robot shares its decision with its peers and starts moving toward the planned waypoint. During its movement, each robot takes location-tagged signal measurements (observations) at a fixed sampling rate. Once the robot reaches the planned waypoint, it runs a check for ending the mission (based on algorithm
termination criteria), and if unsatisfied, it proceeds to decide its next waypoint. This generic planning process involves four major sub-steps, represented by the four blocks inside the “Waypoint Planning” block in Fig. 1. First, the robot combines its own recent observations with the recent observations received (over a wireless network) from its peers, and then down-samples the new data set for preserving the tractability of onboard updating of the GP model. Subsequently, it uses the new data set to update its GP model. Subsequently, it uses the new data set to update its GP model of the environment would represent the greedy (exploitation) choice at each waypoint planning instance. Fig. 1. Bayes-Swarm architecture for each robot in the swarm.

### 3.2 Acquisition Function.

For swarm robotic search, it is important to design an acquisition function that accounts for the characteristics of the embodied search process, i.e., where (i) data are collected and uncertainty is reduced over trajectories and not over separate points in the domain of interest and (ii) robots can only travel finite distances constrained by their maximum speed over a given time-step. Here, we design our own acquisition function partly motivated by the work of Morere et al. [30]. However, in the future, there remains opportunity to translate other well-known acquisition functions such as GP-upper confidence bound (GP-UCB) [31] and q-expected improvement (q-EI) [32] from the Bayesian optimization (BO) and batch-BO domain [23] to suit the needs of swarm robotic search. Below, we describe our unique design of the acquisition function, and how it is used by each robot for waypoint planning.

Robot-\(r\) solves an optimization problem based on its information set \((D_{kr}^{[x]} \text{ and } X_{kr}^{[x]})\), including self-observations and shared peers’ observation from the beginning of the mission till the decision-time \(k_{-}(D_{kr}^{[x]} = \bigcup_{p=1}^{P} \bigcup_{i=1}^{I_{p}} D_{r}^{[y]} ; P_{r} = [X_{r}^{[x]}, y_{r}^{[y]}])\) and the current local peers’ next waypoint \((X_{cr}^{[x]} = \bigcup_{p=1}^{P_{cr}} \bigcup_{i=1}^{I_{cr}} X_{cr}^{[x]})\). For the \(r\)th robot, our mathematical formulation of the acquisition function can be expressed as

\[
x_{r}^{[x]+1} = \arg \max_{X_{kr}^{[x]}} \left( \alpha \cdot h_{r}(x, D_{kr}^{[x]}) + (1 - \alpha)g_{r}(x, D_{kr}^{[x]}, X_{cr}^{[x]}) \right)
\]

subject to

\[
0 \leq t_{r}^{[r]} = \|x - x_{r}^{[x]}\| \leq VT
\]

where the first term, \(h_{r}()\), leads robot \(r\) to the expected location of the source (exploitation) and the second term, \(g_{r}()\), minimizes the knowledge uncertainty of robot \(r\). The coefficient \(\alpha \in [0, 1]\) represents the exploitation weight, i.e., \(\alpha = 1\) would lead to purely exploitative behavior. The length \((l)\) of the path \(s\) is bounded based on the decision-horizon \(T\) and the nominal velocity of the robots \((V)\). The individual terms of the acquisition function are described next.

### Algorithm 1

The Bayes-Swarm algorithm

**Input:** \(G_{kr}, x_{kr}^{[x]}, X_{kr}^{[x]}\)—the current location and recent observations of the robot \((x)\), and the next waypoints of its peers \((X_{cr}^{[x]})\).

**Output:** \(x_{r}^{[x]+1}\)—the next waypoint of robot-\(r\) at its iteration \(k_{r}\).

1. procedure takeDecision\((r, k_{r}, N_{r}, \Delta 0)\)
2. if \(k_{r} = 0\) then
3. \(x_{r}^{[x]} \leftarrow \text{takeFirstDecision}(r, k_{r}, N_{r}, \Delta 0)\)
4. else
5. if Size of \(D_{kr}^{[x]} > N_{max}\) then \(\triangleright N_{max} = 400\)
6. \(\text{Down-sample } D_{kr}^{[x]} \text{ to } N_{max} \text{ observations}\)
7. \(x_{r}^{[x]} \leftarrow \text{by solving the optimization, Eq. (8)}\)
8. \(k_{r} \leftarrow k_{r} + 1\)
9. return \(x_{r}^{[x]+1}, k_{r}\)
10. procedure takeFirstDecision\((r, N_{r}, \Delta 0, V, T)\)
11. \(d \leftarrow VT\)
12. if \(\Delta 0 = 360\) then \(\triangleright \Delta 0\): Initial feasible direction range
13. \(0 \leftarrow r\Delta 0/N_{r}\)
14. else
15. \(0 \leftarrow r\Delta 0/(N_{r} + 1)\)
16. \(x_{r}^{[x]} \leftarrow [d \cos \theta, d \sin \theta]\)
17. return \(x_{r}^{[x]}\)

### 3.3 Source Seeking Formulation

The source location is the optimum point of the source signal. In this approach, robots model the source signal using a GP and the location with the maximum expected value based on their then-current GP model of the environment would represent the greedy (exploitative) choice at each waypoint planning instance. Due to the motion constraints of the robot and limited decision-time horizon, all such locations may not be a feasible choice. To consider this factor, we define the source seeking term as follows:

\[
h_{s}(x, D) = \frac{1}{1 + \|x - X_{r}^{[x]}\|^{2} + \|x - X_{cr}^{[x]}\|^{2}}
\]

where \(X_{r}^{[x]} = \arg \max_{X_{kr}^{[x]}} \mu_{kr}(X)\).

### 3.4 Knowledge-Gain Formulation

As we mentioned in the first section, robots typically gather information over their path; therefore, different paths cause different knowledge-gains. This concept is known as informative path planning, where robots plan paths such that best/maximum possible information is extracted. In this paper, we are interested in paths that minimizes the uncertainty in the robots’ belief (knowledge), which is analogous to maximizing the knowledge-gain. For this purpose, we are estimating the uncertainty in the belief (modeled by a GP) based on the gathered observations and the planned future observations (other robots’ planned paths). We thus define
the knowledge-gain as follows:

\[ g(x, D, \hat{X}) = \int_{\Omega_{\hat{X}}} \sigma(s(u)) \, du \]  

(11)

where the path is written in the parametric form as

\[ s(u) = u\hat{x} + (1 - u)x_k^r; \quad u \in [0, 1] \]  

(12)

3.5 Information Sharing

Inter-robot communication is a key element of any swarm system, and robots often require to communicate with each other over an ad hoc wireless network in outdoor applications. However, given the bandwidth limitations of ad hoc wireless communication and the energy footprint of wireless communication [33], it is typically desirable to reduce the communication burden. To this end, in our proposed method, the decision-making is allowed to be asynchronous and robots share only a down-sampled set of observations. Table 1 provides a quick overview of the type and frequency of the information shared by each robot with all its peers across the swarm. Algorithm 2 lists two procedures that each robot uses to share or receive information. Robots then proceed to individually update their respective knowledge model based on their own information and the future plan of its peers. Having presented an overview of the Bayes-Swarm method, the next section introduces its distributed virtual implementation, case studies developed to test the performance of Bayes-Swarm, and the corresponding implementation settings that we used.

**Algorithm 2** Communication procedures

1: procedure receiveInformation \( r, p, x_k^p, D_k^p \)
2: \( D_k^{i,k} \leftarrow D_k^{i,k} \cup D_k^p \)
3: \( x_k^{i,j}(1:2) \leftarrow x_k^{i,j}(3:4) \)
4: \( x_k^{i,j}(3:4) \leftarrow x_k^{i,j}(1:2) \)
5: return \( D_k^{i,k}, x_k^{i,j} \)
6: procedure sendInformation \( r, x_k^r, D_k^r \)
7: if \( k < 0 \) then
8: Broadcast \( x_k^r \) \( \triangleright 4 \) bytes
9: else
10: Broadcast \( [x_k^r; D_k^r] \) \( \triangleright 4 + 6T \) bytes

3.6 Downsampling Collective Swarm Observations

In order to keep the Bayes-Swarm algorithm scalable and computationally tangible, we are required to downsample the collective data set of the swarm (i.e., observations made by the agents). This is because updating the GP models presents a cubic time complexity \( O(n^3) \) with respect to the size \( n \) of the data set. In this work, we use a simple downsampling approach, known as sample rate compression by an integer factor \( M \) [34]. This approach reduces the data set by keeping the first sample and then every \( M \)-th sample after the first, where \( M = \lceil \text{size}(D^{i,k})/N_{\text{max}} \rceil \).

**Table 1** Content, size, and frequency of information shared by robot \( r \) via communication across the swarm

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inter-robot communication frequency</td>
<td>After each waypoint planning instance</td>
</tr>
<tr>
<td>Content of transmitted data</td>
<td>(1) Its next location to visit ( x_k^r ) and (2) its observations over the last path ( D_k^r )</td>
</tr>
<tr>
<td>Average size of outgoing packets (with time-horizon 1 min)</td>
<td>364 bytes</td>
</tr>
</tbody>
</table>

4 Numerical Experiments and Case Studies

4.1 Distributed Virtual Implementation of Bayes-Swarm.

In order to enable a better representation of the distributed planning process embodied by a physical swarm of robots, we develop a simulated environment that provisions a parallel computing deployment of Bayes-Swarm. This uses “MATLAB”’s parallel programming capabilities to invoke 40 dedicated threads. Each robot operates (the behavior illustrated in Fig. 1) in parallel with respect to the rest of the swarm, updating its own knowledge model after each waypoint and deciding its own next waypoint. The entire process is simulated in a virtual environment developed with MATLAB R2017b and is executed on a workstation with Intel® Xeon Gold 6148 27.5M Cache 2.40 GHz, 20 cores processor and 196 GB RAM. The simulation time-step is set at 1 ms. Robot settings: we set the velocity of each swarm robot at 10 cm/s based on the specifications of e-puck 2 [35]. The observation frequency is set at 1 Hz. To keep the computational complexity of refitting the GP low, the size of data \( (D^{i,k}) \) used by each robot is downsized to 400 (i.e., when it grows beyond 400 in the latter stages of the mission).

4.2 Case Studies.

We design and execute a set of numerical experiments to investigate the performance of the proposed decentralized Bayes-Swarm approach. In order to provide an insightful understanding of the Bayes-Swarm algorithm, three types of tests are conducted for all case studies and the results are evaluated and compared in terms of completion time, cost incurred by robots, knowledge-gain per robot, and mapping error. Mapping error measures how the estimated response surface using GP deviates from the actual response surface of the source in terms of the root-mean-square-error metric. The three tests are described next. Experiment 1: a parametric analysis is conducted to study how the exploitation coefficient of Bayes-Swarm affects its performance. Experiment 2: a scalability analysis is conducted to investigate the performance of Bayes-Swarm across multiple swarm sizes. Experiment 3: Bayes-Swarm is run using the default values listed in Table 2 to analyze its performance in response to different single- and multi-modal spatial distribution of signal strength, and results are compared with those of standard exhaustive search and random walk methods. Experiment 4: The performance of Bayes-Swarm is compared with that of glowworm optimization algorithm proposed by Krishnaman et al. [18], tested on case study 5.

To conduct the first three experiments stated above, five distinct case studies are defined, each corresponding to a different spatial distribution of the signal strength, as shown in Fig. 2. The first case study is a large convex source signal distribution and the rest of the case studies are non-convex multi-modal signal distributions (involving multiple signal sources). Case study 4 is expected to be the most challenging case as it contains one global maxima (target source) and five local maxima (weaker sources) in a large area. Case study 5, adopted from Ref. [36], contains one global maxima (target source) and two local maxima (weaker sources).

In this paper, Bayes-Swarm utilizes two termination criteria during operation. The primary criterion terminates the search if any robot arrives within \( \epsilon \)-vicinity of the source signal location. In addition, Bayes-Swarm terminates if the operation reaches a maximum allowed search time \( (T_{\text{max}}) \). The distance threshold \( \epsilon \) is set at 5 cm and the maximum search time \( T_{\text{max}} \) is outlined for

**Table 2** Maximum allowed search time, \( T_{\text{max}} \) (in s), for case studies

<table>
<thead>
<tr>
<th>Case study</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayes-Swarm</td>
<td>500</td>
<td>100</td>
<td>500</td>
<td>700</td>
<td>100</td>
</tr>
<tr>
<td>Random-walk</td>
<td>4000</td>
<td>50,000</td>
<td>60,000</td>
<td>60,000</td>
<td>10,000</td>
</tr>
</tbody>
</table>
each case study in Table 2. The decision-time horizon ($T$) is set at 4 s for the first decision-making step; then it changes to 10 s for the later decision-making steps.

4.3 Demonstrating Bayes-Swarm: Case Study 2. Figure 3 depicts four snapshots of the Bayes-Swarm algorithm for case study 2 with four robots and $\alpha = 0.4$. It can be seen from this figure that the estimated knowledge model and its uncertainty improves by exploring the search space. The top figures show the uncertainty map ($\sigma(x)$) and the bottom figures show the robot location and its knowledge state (dashed contours). In the bottom figures, the gray solid contours represent the actual source signal (ground truth) and the gray dashed contours represent the source signal (knowledge) model of a robot at the stated time point. Blue solid lines show the paths that robots have already traveled and the observations over which have been shared with all peers, assisting the refitting of their knowledge model. The red solid line shows the paths traveled but the observations over which have not yet been shared with peers. The red dashed lines represent the paths that have been planned but not yet traveled. (Color version online.)
signal (knowledge) model of a robot at the stated time point. Blue solid lines show the paths that robots have already traveled and the observations over which have been shared with all peers, assisting the refitting of their knowledge model. The red solid line shows the paths traveled but the observations over which have not yet been shared with peers. The red dashed lines represent the paths that have been planned but not yet traveled.

From Figs. 3(a)–3(e), it can be seen that when robot 1 reaches its first waypoint, only four self observations are available to it; hence it is able to build only a relatively inaccurate knowledge model (that gives the expected location of the source at 1, 1, 0), which is in reality far away from both of the actual sources. When the last robot (robot 4) takes decision, it has its peers’ observations at \( t = 4^* \) s. The knowledge model (Fig. 3(f)) is still inaccurate, but the uncertainty map (Fig. 3(b)) is improved. After 26 s (Figs. 3(c) and 3(g)), the robots are able to converge to a fairly accurate knowledge model of the signal distribution, and their future updates and planning (seen in Figs. 3(d) and 3(h)) puts two robots in the team within the threshold of the source location at time \( t = 54 \) s.

5 Results and Discussion

5.1 Experiment 1: Parameter Analysis of Bayes-Swarm. In the proposed decentralized method, there is one major prescribed parameter that needs to be prescribed or tuned—the exploitation coefficient parameter \( \alpha \)—that regulates the balance between exploration and exploitation. We run an experiment to study how this exploitation coefficient parameter (\( \alpha \) varying from 0 to 1) affects the performance of Bayes-Swarm for the case studies 2 and 4, across multiple swarm sizes. Snapshots of the final state of robots for three values of \( \alpha \) for the case study 2 with four robots are depicted in Fig. 4. The performance outcomes in terms of completion time, and mapping error are summarized in Figs. 5 and 6.

Pure source seeking (\( \alpha = 1 \)): One of the extreme case happens when the knowledge-gain term is eliminated in the objective function; in this mode, robots try to reach the expected source location faster without exploring the area (getting enough knowledge)—basically the purely greedy approach. For this purpose, the exploitation coefficient is set at \( \alpha = 1 \). Figure 4(c) illustrates the behavior of robots under this setting. It can be seen from this figure that the estimated source signal or knowledge model is quite inaccurate due to the lack of explorative search.

Only knowledge-gain term (\( \alpha = 0 \)): By setting \( \alpha = 0 \), the objective function (Eq. (8)) is reduced to the knowledge-gain term (Eq. (11)), which results in purely explorative search. As expected, under this setting, robots are able to estimate a relatively accurate model of signal distribution (Fig. 4(a)). This mode is suited for mapping applications, such as mapping offshore oil spills [12].

Combined source seeking and knowledge-gain terms—different trade-offs (0 < \( \alpha \) < 1): By setting the exploitation coefficient \( \alpha \) at values between 0 and 1, we can tune the degree of exploration and exploitation of the swarm search. Figures 5(b) and 6(b) show that by increasing the exploitation coefficient from 0 to 1, the mapping error increases, especially for \( \alpha \) values beyond 0.3. Figure 4(b) depicts the search behavior of the swarm for \( \alpha = 0.4 \). In this setting,
one robot successfully reaches the source location while other robots are still exploring the search area. Depending on the complexity of the source signal distribution, the effect of exploitation coefficient parameter on the estimation of the knowledge model will vary.

In terms of completion time, the complexity of the source signal distribution and the initial path of robots play important roles. In case study 2, the impact of $\alpha$ on completion time varies with the size of the robot team (Fig. 5(a)). In case study 4, we can see from Fig. 6(a) that Bayes-Swarm with $\alpha > 0.04$ is not able to lead the robots to find the target/source within the maximum allowed time (700 s). In order to get the best performance, the exploitation coefficient ($\alpha$) needs to be less than 0.02. This is attributed to the need for greater exploration in a multimodal environment. In summary, for choosing the correct value of $\alpha$ to get the best performance, we need to consider the number of robots, the complexity of the source signal distributions, and the robots’ capabilities.

5.2 Experiment 2: Scalability Analysis of Bayes-Swarm. In this experiment, we use case study 4 to perform an analysis of how the size of the robot swarm impacts Bayes-Swarm’s performance. To this end, we run Bayes-Swarm simulations with $\alpha = 0.4$ and swarm sizes varying from 2 to 100. Figure 7 illustrates the results of this analysis in terms of the completion time, averaged knowledge-gain of each robot ($\bar{g}(s)$), averaged number of decisions per robot ($\bar{Nd}$), and mapping error. The results show that the performance improves by increasing the size of the swarm from 2 to 100, with completion time reducing by ~41.3%. Moreover, the averaged number of decisions (waypoint planning instances) per robot and the averaged knowledge-gain per robot, respectively, decrease by about 64% and 83.3% when the swarm size grows from 2 to 100. Although the mapping error with 100 robots is 16.6% less than the mapping error with 2 robots, increasing the number of robots does not universally improve the mapping error, as evident from the non-monotonic trend seen in the top right plot of Fig. 7 (unless $\alpha$ is tuned based on the size of swarm).

To summarize the observations made from Fig. 7, increasing the size of swarms becomes increasingly effective for complex signal distribution environments. However, beyond a certain swarm size (~20 in this analysis), there is a decreasing rate of improvement. These observations provide strong evidence of the scalability of the Bayes-Swarm method. At the same time, they highlight the importance of identifying suitable team sizes for suitable mission profiles, given resource constraints and time sensitivity of the mission.

5.3 Experiment 3: Comparative Analysis With Baselines. Exhaustive search and random-walk algorithms are implemented along with Bayes-Swarm for comparative analysis. We test these algorithms to find the source location in the five case studies, illustrated in Fig. 2. The settings of Bayes-Swarm are not individually tuned for each case, in order to allow fair comparison; the exploitation coefficient is set at 0.4 and $T$ at 4 s. Table 3 summarizes the results of this experiment in terms of the completion time. In this experiment, the maximum allowed search time for random-walk is adjusted to 1.5 times of what is needed by exhaustive search for each case study environment. In case study 4, we partition the arena into four parts and each robot searches one part using the exhaustive search method. Note that, in this table, we only report the best performance across five runs of the random-walk method for each case.

The results show that the Bayes-Swarm algorithm performs significantly better than the exhaustive search and random-walk
Bayes-Swarm with a state-of-the-art method, the well-known Intelligence Method. To perform further comparative analysis of case studies 1 and 2, we compare Bayes-Swarm and the glowworm algorithm in two case studies, while the glowworm algorithm is designed to find the source with maximum strength in the presence of other weaker (say decoy) sources, while the glowworm algorithm is designed to find the source with maximum signal strength in 2.43, 3.32, and 2.40 time units for \( \alpha < 0.4 \) works well. Hence, we explored how Bayes-Swarm would compare to the glowworm algorithm when implemented with different values of \( \alpha \). The results show that Bayes-Swarm found the target with maximum strength in 2.43, 3.32, and 2.40 time units for \( \alpha \) set at 0.05, 0.1, and 0.2, respectively; these mission completion times are still better than that resulting from the glowworm implementation.

5.5 Discussion of Bayes-Swarm Performance. The various empirical analyzes performed here show that the proposed Bayes-Swarm algorithm is scalable with respect to the number of robots and is able to localize targets involving complex multimodal signal environments. The algorithm also requires minimal heuristics, being dependent on only a single tunable parameter \( \alpha \) that regulates the balance between exploration and exploitation. While, in the numerical experiments presented in this paper, a value of \( \alpha < 0.4 \) was found to yield promising performance, in future it would be important to pursue approaches to automatically adapt \( \alpha \) to the environment during the mission. Another important advantage of the Bayes-Swarm algorithm over model-free algorithms is its provision of a belief model of the signal environment during the mission. This makes the emergence of the robotic swarm system, from agent level micro-planning to team level macro search dynamics, relatively interpretable (as opposed to a blackbox phenomena).

The current form of the Bayes-Swarm method needs to overcome a few crucial limitations in the future. The need for down sampling is one of them. Since updating the GP-based belief model presents a cubic time complexity with respect to data-set size, we are required to downsample the collective data set of the swarm (observations made by agents). Without such down sampling, the cost of updating the belief model and thus of decentralized waypoint planning onboard simple robotic agents (with frugal computing capacity) will become burdensome. Currently, we use a simple down sampling approach based on sample rate compression. While this approach is simple to implement, it is far from optimal, especially for large swarm systems with 100s of robots. In such scenarios, choosing the most informative samples (to update the belief model) will need to be addressed in future research on Bayes-Swarm. Another limitation of the current approach is the assumption of full observability, or a fully connected wireless network, where each swarm-robotic agent can communicate with all team members. In practice, it is more common to experience

<table>
<thead>
<tr>
<th>Case study</th>
<th>Algorithm</th>
<th>Total time ( a ) (s)</th>
<th>Success rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bayes-Swarm</td>
<td>246.1</td>
<td>1/1</td>
</tr>
<tr>
<td></td>
<td>Random-walk</td>
<td>20,394</td>
<td>1/5</td>
</tr>
<tr>
<td></td>
<td>Exhaustive search</td>
<td>22,174</td>
<td>1/1</td>
</tr>
<tr>
<td>2</td>
<td>Bayes-Swarm</td>
<td>42.5</td>
<td>1/1</td>
</tr>
<tr>
<td></td>
<td>Random-walk</td>
<td>227.6</td>
<td>5/5</td>
</tr>
<tr>
<td></td>
<td>Exhaustive search</td>
<td>225.3</td>
<td>1/1</td>
</tr>
<tr>
<td>3</td>
<td>Bayes-Swarm</td>
<td>260.1</td>
<td>1/1</td>
</tr>
<tr>
<td></td>
<td>Random-walk</td>
<td>–</td>
<td>0/5</td>
</tr>
<tr>
<td></td>
<td>Exhaustive search</td>
<td>22,174</td>
<td>1/1</td>
</tr>
<tr>
<td>4</td>
<td>Bayes-Swarm</td>
<td>373.2</td>
<td>1/1</td>
</tr>
<tr>
<td></td>
<td>Random-walk</td>
<td>–</td>
<td>0/5</td>
</tr>
<tr>
<td></td>
<td>Exhaustive search</td>
<td>916b</td>
<td>1/1</td>
</tr>
<tr>
<td>5</td>
<td>Bayes-Swarm</td>
<td>31.9</td>
<td>1/1</td>
</tr>
<tr>
<td></td>
<td>Random-walk</td>
<td>–</td>
<td>0/5</td>
</tr>
<tr>
<td></td>
<td>Exhaustive search</td>
<td>992b</td>
<td>1/1</td>
</tr>
</tbody>
</table>

\(^a\)As all random-walk runs are not able to find the source, we only report the total time of the best solution obtained using the random-walk. \(^b\)For this case, we divide the search space into four equal quarters and each robot does an exhaustive search in each portion (two in the global portion).
Bayes-Swarm allows us to more comprehensively explore the scalability of the acquisition function (sharing of data), especially for applications where 100s of observations history of robot-\(r\), including self-observations and shared by its peers, from beginning of the mission until finishing its \(r\)th waypoint.

\[ x^{t+1}_{r} = \text{next waypoint of robot-}r \text{ at the decision-time } k_{r} \]

\[ y_{r}' = \text{source signal measurements made by robot-}r \text{ while it is moving from waypoint-}(i-1) \text{ to waypoint-}i \]

\[ D_{r} = \text{a set of observations of an environment that made by robot-}r \text{ after finishing its } r \text{th waypoint; i.e., } D_{r} = [X_{r}', Y_{r}'] \]

\[ x_{r}^{'}, y_{r}^{'}, \alpha = \text{current local peer-p’s next waypoint of robot-}r \text{ at the decision-time } k_{r} \]

\[ h_{r}(\cdot) = \text{source seeking term of robot-}r \text{ in Bayes-Swarm} \]

\[ g_{r}(\cdot) = \text{knowledge-gain term of robot-}r \text{ in Bayes-Swarm} \]

\[ GP_{r} = \text{GP model trained and used by robot-}r \]

\[ \alpha = \text{exploitation weight, where } \alpha = 1 \text{ would be purely exploitative} \]

\[ \Delta 0 = \text{initial feasible direction} \]

**Appendix A: Definition of Case Studies**

**Case Study 1: Large Arena, Convex Signal Distribution**

\[ f = \exp \left( -\frac{\| x - c_{1} \|^{2}}{130} \right) \]  

(A1)

Here, \( x = (x_{1}, x_{2}) \), where \( 0 \leq x_{1} \leq 24 \) and \( c_{1} = (5, 23) \). The initial feasible direction, \( \Delta 0 \), is set at 90.

**Case Study 2: Small Arena, Non-Convex Signal Distribution**

\[ f = \exp \left( -\frac{\| x - c_{1} \|^{2}}{3} \right) + \frac{1}{2} \exp \left( -2\| x - c_{2} \|^{2} \right) \]  

(A2)

Here, \( x = (x_{1}, x_{2}) \), where \( 0 \leq x_{1} \leq 2.4 \) and \( c_{1} = (1.9, 2.3) \), and \( c_{2} = (1.5, 0.5) \). The initial feasible direction, \( \Delta 0 \), is set at 90.

**Case Study 3: Large Arena, Non-Convex Signal Distribution**

\[ f = \exp \left( -\frac{\| x - c_{1} \|^{2}}{120} \right) + \frac{1}{2} \exp \left( -\frac{\| x - c_{2} \|^{2}}{30} \right) \]  

(A3)

Here, \( x = (x_{1}, x_{2}) \), where \( 0 \leq x_{1} \leq 24 \), \( c_{1} = (10, 23) \), and \( c_{2} = (15, 5) \). The initial feasible direction, \( \Delta 0 \), is set at 90.

**Case Study 4: Large Arena, Highly Multi-Modal Signal Distribution**

\[ f = \exp \left( -\frac{\| x - c_{1} \|^{2}}{130} \right) + \frac{2}{5} \sum_{i=2}^{4} \exp \left( -\frac{\| x - c_{i} \|^{2}}{40} \right) \]  

(A4)

Here, \( x = (x_{1}, x_{2}) \), where \(-24 \leq x \leq 24 \). Moreover, \( c_{1} = (21, 19) \), \( c_{2} = (21, -19) \), \( c_{3} = (0, -15) \), \( c_{4} = (0, 15) \), \( c_{5} = (-19, 10) \), \( c_{6} = (21, 19) \), and \( c_{7} = (-15, -15) \). The initial feasible direction, \( \Delta 0 \), is set at 360.

**Acknowledgment**

Support from the National Science Foundation Award IIS-1927462 is gratefully acknowledged. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF.

**Nomenclature**

- \( r \) = robot index, a value between 0 and \( N_{r} \)
- \( T \) = decision-horizon time of robots
- \( V \) = nominal velocity of robots
- \( l_{s} \) = length of path \( s \)
- \( N_{\text{max}} \) = downsample threshold, which defines the maximum allowed samples for fitting the GP model by each robot
- \( N_{r} \) = number of robots (swarm size)
- \( D_{r}^{k} \) = observations history of robot-\( r \), including self-observations and shared by its peers, from beginning of the mission until finishing its \( r \)th waypoint
- \( x_{r} \) = next waypoint of robot-\( r \) at the decision-time \( k_{r} \)
- \( y_{r}' \) = source signal measurements made by robot-\( r \) while it is moving from waypoint-\((i-1)\) to waypoint-\( i \)
- \( D_{r} = \) a set of observations of an environment that made by robot-\( r \) after finishing its \( r \)th waypoint; i.e., \( D_{r} = [X_{r}', Y_{r}'] \)
- \( x_{r}^{'}, y_{r}^{'} \) = current local peer-\( p \)'s next waypoint of robot-\( r \) at the decision-time \( k_{r} \); i.e., \( X_{r} = \bigcup_{i=1}^{N_{r}} X_{r}^{p} \)
- \( h_{r}(\cdot) \) = source seeking term of robot-\( r \) in Bayes-Swarm
- \( g_{r}(\cdot) \) = knowledge-gain term of robot-\( r \) in Bayes-Swarm
- \( GP_{r} \) = GP model trained and used by robot-\( r \)
- \( \alpha \) = exploitation weight, where \( \alpha = 1 \) would be purely exploitative
- \( \Delta 0 \) = initial feasible direction
The function consists of a set of three peaks at locations given in Ref. [36], which is a multi-modal function and denotes enhancement constant; decision range gain; sensor range of robots; distance moved by each glowworm when a decision is taken.

Case Study 5: Multi-Modal Source. We use the first example given in Ref. [36], which is a multi-modal function and defined as follows:

\[
f(x_1, x_2) = 3(1 - x_1^2) \exp(-x_1^2) - (x_2 + 1)^2 \\
-10(x_1/5 - x_1^3 - x_1^5) \exp(-x_1^2 - x_2^2) \\
-1/(3/\Delta s)^2 (x_1 + 1)^2 - x_2^2
\]

Here, $-3 \leq x_1, x_2 \leq 3$. A set of 50 robots are randomly deployed in a two-dimensional region such that $-3 \leq x_1 \leq 1.2$ and $-3 \leq x_2 \leq 3$. The function consists of a set of three peaks at locations $(-0.0093, 1.5814)$, $(1.2857, -0.0048)$, and $(-0.46, -0.6292)$. The source with maximum strength is located at $(-0.0093, 1.5814)$.

Appendix B: Bayes-Swarm and Glowworm Algorithm Settings

Table 4 summarizes all settings that have been used for Bayes-Swarm for all experiments and case studies. Table 5 lists the settings that have been used for the glowworm algorithm in Experiment 4.

### References


