

An Extended Bayesian Optimization Approach to Decentralized Swarm Robotic Search

Payam Ghassemi

Ph.D. Candidate, Student Member of ASME
Dept. of Mechanical & Aerospace Eng.
University at Buffalo
Buffalo, NY, 14260
Email: payamgha@buffalo.edu

Souma Chowdhury*

Assistant Professor, Professional Member of ASME
Dept. of Mechanical & Aerospace Eng.
University at Buffalo
Buffalo, NY, 14260
Email: soumacho@buffalo.edu

Abstract: *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. Favor-*

able scalability characteristics are also demonstrated.

Keywords: Swarm Robotic Search, Informative Path Planning, Bayesian Optimization, Gaussian Process.

Nomenclature

N_r : Number of robots (swarm size)

r : Robot index, a value between 0 and N_r

\mathbf{X}_r^i : The location of the observations made by robot- r while it is moving from waypoint- $(i-1)$ to waypoint- i

\mathbf{y}_r^i : The source signal measurements made by robot- r while it is moving from waypoint- $(i-1)$ to waypoint- i

\mathcal{D}_r^i : A set of observations of an environment that made by robot- r after finishing its i -th waypoint; i.e., $\mathcal{D}_r^i = [\mathbf{X}_r^i, \mathbf{y}_r^i]$

$\mathcal{D}^{1:k_r}$: The observations history of robot- r , including self-observations and shared by its peers, from beginning of the mission until finishing its i -th waypoint

$\hat{\mathbf{X}}_{-rp}^{k_r}$: The current local peer- p 's next waypoint of robot- r at the decision-time k_r

$\hat{\mathbf{X}}_{-r}^{k_r}$: The current local peers' next waypoint of robot- r at the decision-time k_r ; i.e., $\hat{\mathbf{X}}_{-r}^{k_r} = \bigcup_{p=1; p \neq r} \hat{\mathbf{X}}_{-rp}^{k_r}$

\mathbf{x}_r^{k+1} : The next waypoint of robot- r at the decision-time k_r

$h_r(\cdot)$: The source seeking term of robot- r in Bayes-Swarm

$g_r(\cdot)$, The knowledge-gain term of robot- r in Bayes-Swarm

α : The exploitation weight, where $\alpha = 1$ would be purely exploitative.

l_s : The length of path s

T : The decision-horizon time of robots

*Address all correspondence to this author.

V : The nominal velocity of robots
 GP_r : The Gaussian process (GP) model trained and used by robot- r
 $\Delta\theta$: Initial feasible direction
 N_{\max} : Downsample threshold, which defines the maximum allowed samples for fitting the GP model by each robot

1 Introduction

Swarm robotic search is concerned with searching for or localizing targets in unknown environments with a large number of collaborative 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 light-weight (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 [13], model-driven strategies, [14], Bayesian filter by in-

corporating mutual information [15], strategies based on local cues [16], and uncertainty reduction methods [17]. Scaling these methods from the multi-robotic (<10 agents [11]) to the swarm-robotic level (10 to 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 that based on nature-inspired *swarm intelligence* (SI) principles [18–20]. 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 [21, 22]. Majority of the latter methods are targeted at localizing a single source [9, 23]. A notable exception is the Glowworm optimization based algorithm reported by Krishnanand et al. [19]. This approach was showed 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) [23]. 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 [24], 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 \gg 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 [25]. 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 to model the signal distribution knowledge (task inference) and solves 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 [26]¹. 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 [27]. 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 [28]. 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 lies 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

Gaussian process (GP) models [29] are probabilistic surrogate models that have been used successfully in different applications such as modeling the objective function in Bayesian optimization [30]. If we have a set of n observations of an environment, $\mathcal{D} = \mathbf{x}_i, y_i | i = 1 \dots n$, then we can write the following equation by assuming that the observed values y differ from the function $f(\mathbf{x})$ values by an additive noise ε , where \mathbf{x} denotes an input vector:

$$y = f(\mathbf{x}) + \varepsilon \quad (1)$$

By assuming the noise follows an independent, identically distributed Gaussian distribution with zero mean and variance σ_n^2 , we have $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$. The function $f(\mathbf{x})$ can be estimated by a GP with mean function $\mu(\mathbf{x})$ and covariance kernel $\sigma^2(\mathbf{x})$ given by:

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{X}), \sigma^2(\mathbf{X})) \quad (2)$$

where,

$$\mu(\mathbf{x}) = \Lambda(\mathbf{x})(\mathbf{y} - \Phi\beta) \quad (3)$$

$$\sigma^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \Lambda(\mathbf{x})\mathbf{k}_n(\mathbf{x}) \quad (4)$$

$$\Lambda(\mathbf{x}) = \mathbf{k}_n(\mathbf{x})^T [\mathbf{K} + \sigma_n^2(\mathbf{x})\mathbf{I}]^{-1} \quad (5)$$

Here Φ is the vector of explicit basis functions and $\mathbf{K} = \mathbf{K}(\mathbf{X}, \mathbf{X} | \theta)$ is the covariance function matrix such that $(\mathbf{K})_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{k}_n(\mathbf{x}) = [k(\mathbf{x}_1, \mathbf{x}), \dots, k(\mathbf{x}_n, \mathbf{x})]^T$. In this paper, the hyper-parameters of the GP model are optimized by maximizing the log-likelihood P as a function of $\beta, \theta, \sigma_n^2$:

$$\hat{\beta}, \hat{\theta}, \hat{\sigma}_n^2 = \arg \max_{\beta, \theta, \sigma_n^2} \log P(\mathbf{y} | \mathbf{X}, \beta, \theta, \sigma_n^2) \quad (6)$$

where,

$$\begin{aligned} \log P(\mathbf{y} | \mathbf{X}, \beta, \theta, \sigma_n^2) = & -\frac{1}{2}(\mathbf{y} - \Phi\beta)^T \Lambda(\mathbf{x})^{-1}(\mathbf{y} - \Phi\beta) \\ & -\frac{N_s}{2} \log 2\pi - \frac{1}{2} \log |\Lambda(\mathbf{x})| \end{aligned} \quad (7)$$

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 Alg. 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

¹This manuscript is an invited paper, being submitted to the 2019 CIE Research Highlights Issue of the JCISE journal, and is based on our 2019 IDETC/CIE Conference paper numbered IDETC2019-97887, Ref. [26]. JCISE-19-1275, P. Ghassemi & S. Chowdhury

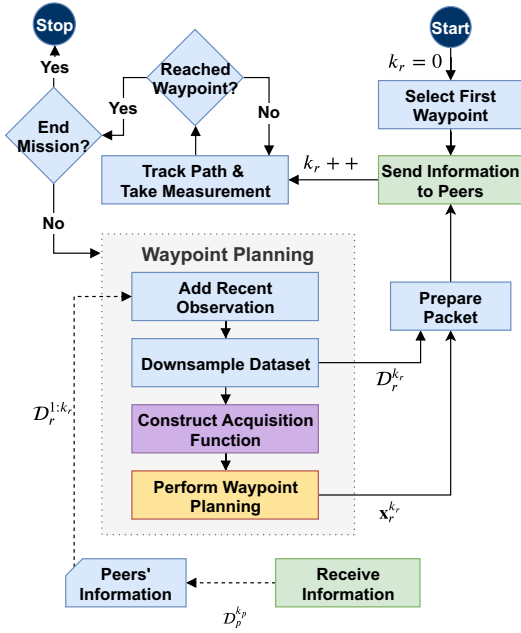


Fig. 1. Bayes-Swarm architecture for each robot in the swarm

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 based acquisition function. The next waypoint is then determined by maximizing this acquisition function, subject to certain range constraints. Each robot then creates an information packet (“Prepare Packet” block in Fig. 1) comprising a downsampled version of its recent observations and its decided next waypoint, and broadcasts that information to all peers, before moving to its next waypoint. This procedure is then repeated until the target of interest is found or other mission termination criteria (e.g., maximum endurance of robots) is reached.

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 is 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. [31]. However, in the future, there remains opportunity to translate other well-known acquisition functions such as GP-UCB [32] and q-EI [33] from the BO and batch-BO domain [24] 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 ($\mathcal{D}^{1:k_r}$ and $\hat{\mathbf{X}}_{-r}^{k_r}$), including self-observations and shared peers’ observation from the beginning of the mission till the decision-time k_r ($\mathcal{D}^{1:k_r} = \bigcup_{i=1}^{N_r} \bigcup_{j=1}^{k_r} \mathcal{D}_r^i$; $\mathcal{D}_r^i = [\mathbf{X}_r^i, \mathbf{y}_r^i]$) and the current local peers’ next waypoint ($\hat{\mathbf{X}}_{-r}^{k_r} = \bigcup_{p=1; p \neq r} \hat{\mathbf{X}}_{-rp}^{k_r}$). For the r^{th} robot, our mathematical formulation of the acquisition function can be expressed as:

$$\mathbf{x}_r^{k+1} = \arg \max_{\mathbf{x} \in \mathcal{X}^{k_r}} \left(\alpha \cdot h_r(\mathbf{x}, \mathcal{D}^{1:k_r}) + (1 - \alpha) g_r(\mathbf{x}, \mathcal{D}^{1:k_r}, \hat{\mathbf{X}}_{-r}^{k_r}) \right) \quad \text{s.t.} \quad (8)$$

$$0 \leq l_s^{k_r} = \|\mathbf{x} - \mathbf{x}_r^{k_r}\| \leq VT \quad (9)$$

where the first term, $h_r(\cdot)$, leads robot r to the expected location of the source (exploitation) and the second term, $g_r(\cdot)$, 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_s) 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.

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 (exploitive) choice at each waypoint planning instance. Due to the motion constraints of the robot and limited decision-time horizon, all such a location may not be a feasible choices. To consider this factor, we define the source seeking term as follows:

$$h_r(\mathbf{x}, \mathcal{D}) = \frac{1}{1 + (\mathbf{x} - \bar{\mathbf{x}}^*)^T (\mathbf{x} - \bar{\mathbf{x}}^*)} \quad (10)$$

where $\bar{\mathbf{x}}^* = \arg \max_{\bar{\mathbf{x}}} \mu(\bar{\mathbf{x}})$.

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 (IPP), 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:

$$g_r(\mathbf{x}, \mathcal{D}, \hat{\mathbf{X}}) = \int_{s(\mathbf{x})} \sigma(s(u)) du \quad (11)$$

where, the path is written in the parametric form as:

$$s(u) = u\mathbf{x} + (1-u)\mathbf{x}_r^{k_r}; u \in [0, 1] \quad (12)$$

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 [34], 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 UAV with all its peers

Algorithm 1 The Bayes-Swarm Algorithm

Input: $GP_r, \mathbf{x}_r, X_{-r}^{k_r}$ - the current location and recent observations of the robot (\mathbf{x}), and the next waypoints of its peers ($X_{-r}^{k_r}$).

Output: $x_r^{k_r+1}$ - the next waypoint of robot- r at its iteration k_r .

```

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

```

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.

3.6 Downsampling Collective Swarm Observations

In order to keep the Bayes-Swarm algorithm scalable and computational 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 [36]. This approach reduces the data set by keeping the first sample and then every M -th sample after the first, where $M = \lceil \text{size}(\mathcal{D}^{1:k_r})/N_{\max} \rceil$.

4 Numerical Experiments & Case Studies

4.1 Distributed Virtual Implementation of Bayes-Swarm

In order to enable a better representation of 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 mil-

Algorithm 2 Communication Procedures

```

1: procedure RECEIVEINFORMATION( $r, p, \mathbf{x}_p^{k_p}, \mathcal{D}_p^{k_p}$ )
2:    $\mathcal{D}_r^{1:k_r} \leftarrow \mathcal{D}_r^{1:k_r} \cup \mathcal{D}_p^{k_p}$ 
3:    $\hat{X}_{-rp}^{k_r}(1:2) \leftarrow \hat{X}_{-rp}^{k_r}(3:4)$ 
4:    $\hat{X}_{-rp}^{k_r}(3:4) \leftarrow \mathbf{x}_p^{k_p}$ 
5:   return  $\mathcal{D}_r^{1:k_r}, \hat{X}_{-rp}^{k_r}$ 
6: procedure SENDINFORMATION( $r, \mathbf{x}_r^{k_r}, \mathcal{D}_r^{k_r}$ )
7:   if  $k_r = 0$  then
8:     Broadcast  $\mathbf{x}_r^{k_r}$   $\triangleright 4$  bytes
9:   else
10:    Broadcast  $\{\mathbf{x}_r^{k_r}; \mathcal{D}_r^{k_r}\}$   $\triangleright 4 + 6T$  bytes

```

Table 1. Content, size, and frequency of information shared by robot r via communication across the swarm.

Property	Descriptions
Inter-robot communication frequency	After each waypoint planning instance
Content of transmitted data	1) Its next location to visit ($\mathbf{x}_r^{k_r}$); and 2) Its observations over the last path ($\mathcal{D}_r^{k_r}$)
Average size of outgoing data packets (with time-horizon 1 min)	364 Bytes

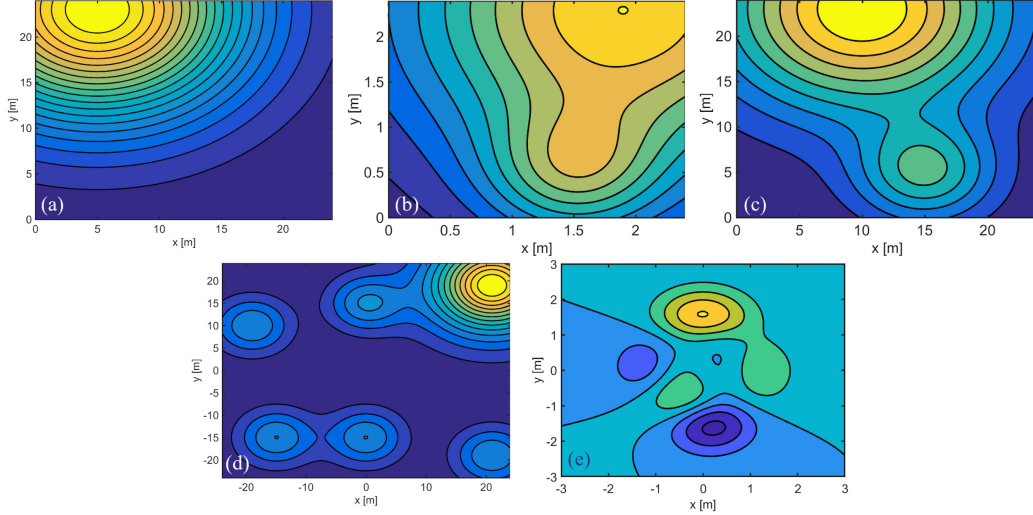


Fig. 2. Five case studies with source distributions of different levels of complexity. (a) Case study 1: large arena, convex signal distribution; (b) Case study 2: small arena, non-convex signal distribution; (c) Case study 3: large arena, non-convex signal distribution; (d) Case study 4: large arena, highly multi-modal signal distribution; (e) Case study 5: small arena, multi-modal signal distribution (given in [35]).

liseconds. *Robot settings:* we set the velocity of each swarm robot at 10 cm/s based on the specifications of e-puck 2 [37]. The observation frequency is set at 1 Hz. To keep the computational complexity of refitting the GP low, the size of data ($\mathcal{D}_r^{1:k_r}$) used by each robot is downsampled 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 (RMSE) 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 per-

formance 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 (GSO) algorithm proposed by Krishnanand et al. [19], 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 arena. Case study 5, adopted from [35], 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 ϵ -vicinity of the source signal location. In addition, *Bayes-Swarm* terminates if the operation reaches a maximum allowed search time (T_{\max}). The distance threshold ϵ is set at 5 cm and the maximum search time T_{\max} is outlined for each case study in Table 2. The decision-time horizon (T) is set at 4 seconds for the first

decision-making step; then it changes to 10 seconds 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 4 robots and $\alpha = 0.4$. It can be seen from this figure how the estimated knowledge model and its uncertainty improves by exploring the search space. The top figures show the uncertainty map ($\sigma(\mathbf{x})$) 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.

From Fig. 3(a)-3(e), it can be seen that when Robot 1 reaches its first waypoint, only 4 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 (1.6, 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 seconds (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 $t = 54s$.

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 α , that regulates the balance between exploration and exploitation. We run an experiment to study how this exploitation coefficient parameter (α 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 α for the case study 2 with 4 robots are depicted in Fig. 4. The performance outcomes in terms of completion time, and mapping error are summarized in Figs. 5-6.

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 (get-

Table 2. Max. allowed search time, T_{\max} (in sec), for case studies.

Case Study	1	2	3	4	5
Bayes-Swarm	500	100	500	700	100
Random-walk	4,000	50,000	60,000	60,000	10,000

ting 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 lacking 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 & knowledge-gain terms – different trade-offs ($0 < \alpha < 1$): By setting the exploitation coefficient α 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 α 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 α 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 seconds). In order to get the best performance, the exploitation coefficient (α) 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 α 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

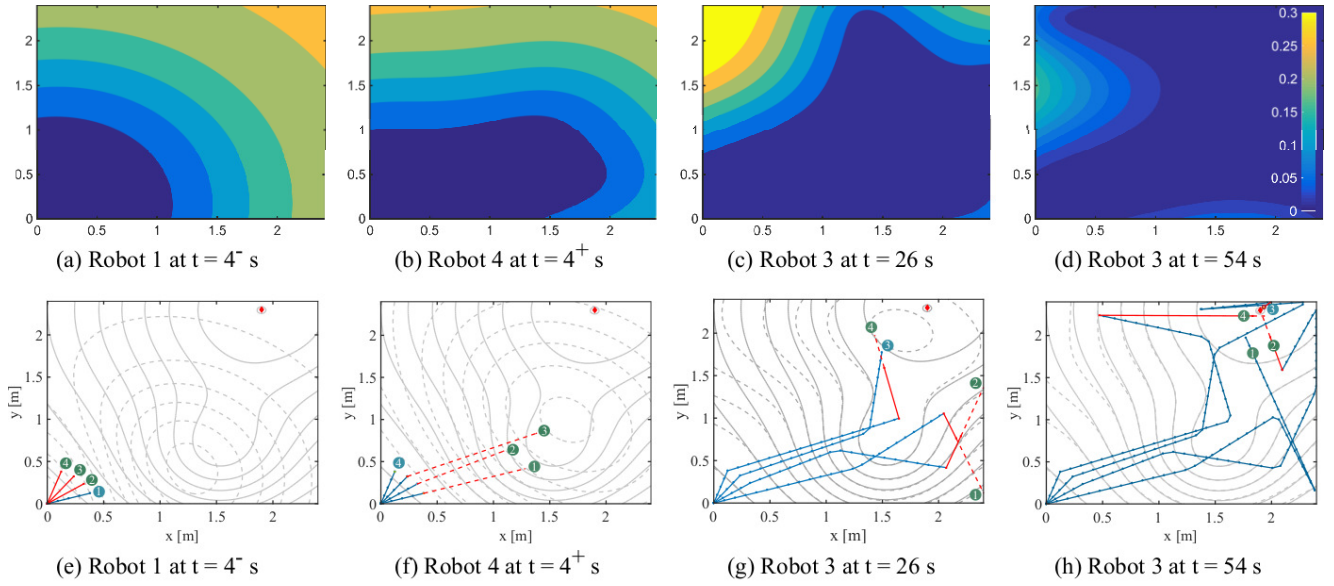


Fig. 3. Snapshots for the case study 2 with 4 robots that run Bayes-Swarm with $\alpha = 0.4$. The top figures show the uncertainty map ($\sigma(\mathbf{x})$) and the bottom figures show the robot and knowledge state. In the bottom figures, the gray solid contours represent the actual source signal (ground 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.

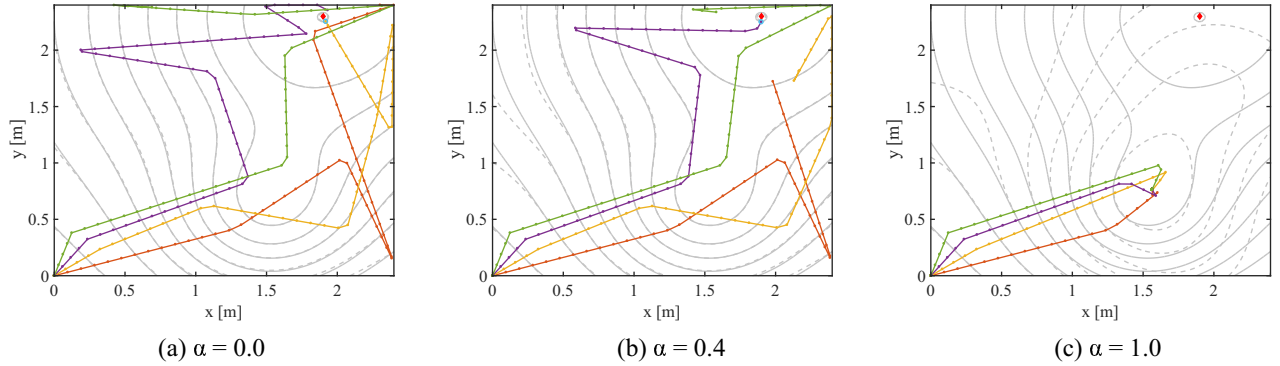
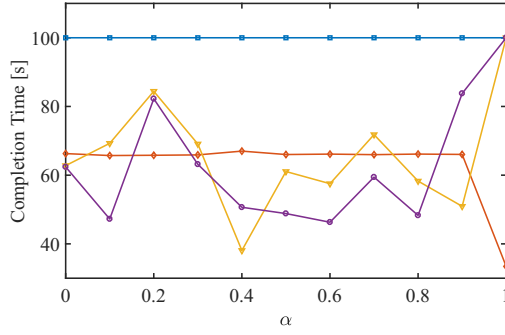


Fig. 4. Performance dynamics of a 4-robot team under different values of the exploitation/exploration balance coefficient (α), for case study 2. The solid line is the actual source signal distribution (ground truth) and the dashed line represents the extracted source signal distribution (knowledge) modeled by the robots. The red dot shows the actual source location.

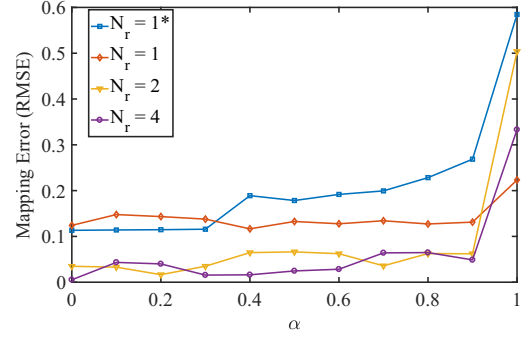
from 2 to 100. Figure 7 illustrates the results of this analysis in terms of the completion time, averaged knowledge-gain of each robot ($\bar{g}(x)$), averaged number of decisions per robot (\bar{N}_d) and mapping error. The results show that the performance improves by increasing the size of the swarm from 2 to 100, with completion time reducing by $\sim 41.3\%$. 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 α is tuned based on the size of swarm).

To summarize the observations made from Fig. 7, increasing the size of swarms become 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.

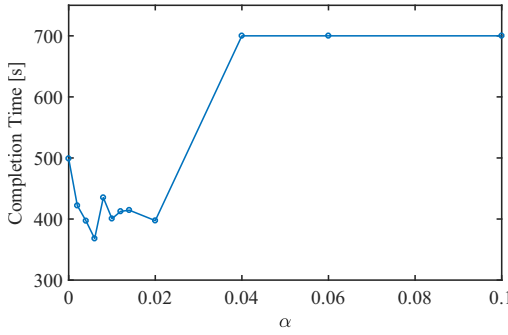


(a) Case study 2: completion time

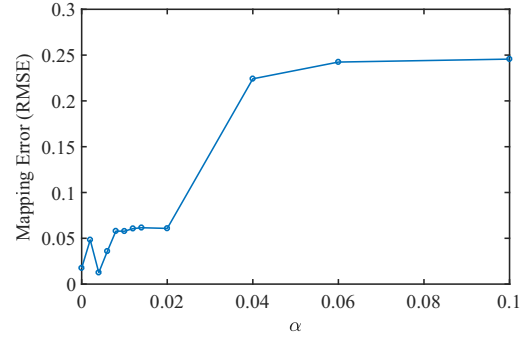


(b) Case study 2: knowledge model accuracy

Fig. 5. Parametric analysis of the exploitation/exploration balance coefficient (α) in case study 2 (small arena, bi-modal signal distribution). For all runs, the maximum allowed search time is set at 100 seconds ($T_{\max} = 100$). In this case study, we provided two 1-robot scenarios, $N_r = 1^*$ and $N_r = 1$, by setting the initial feasible direction ($\Delta\theta$) at 0° and 45° , respectively, to demonstrate the sensitivity of single robot performance on the initial uninformed action.



(a) Case study 4: completion time



(b) Case study 4: knowledge model accuracy

Fig. 6. Performance dynamics of a 10-robot team under different values of the exploitation/exploration balance coefficient (α), case study 4 (large arena, highly multi-modal signal distribution). For all runs, the maximum allowed search time is set at 700 seconds ($T_{\max} = 700$).

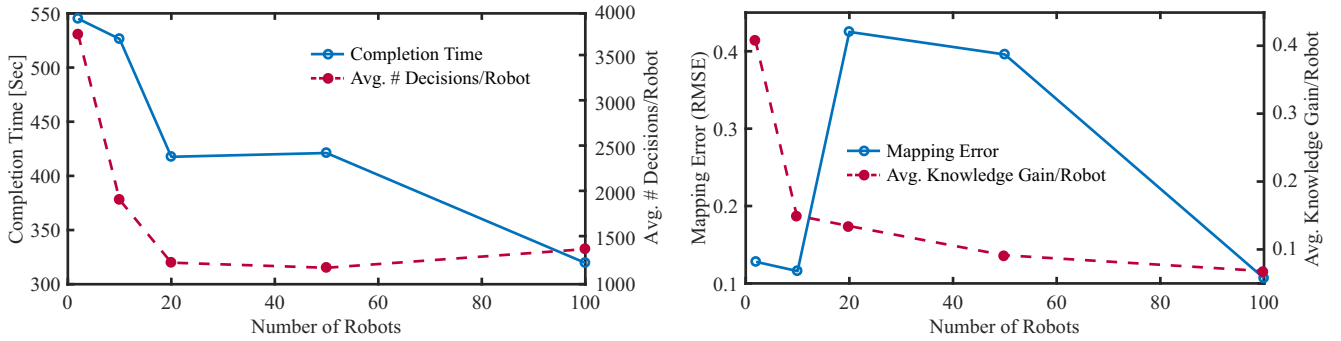


Fig. 7. Scalability analysis of Bayes-Swarm with $\alpha = 0.4$ and swarm sizes varying from 2 to 100, when applied to case study 4 environment.

5.3 Experiment 3: Comparative Analysis w/ 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 seconds. 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 4 parts and each robot searches one part using the exhaustive search method. Note that, in this table, we only report the best performance across 5 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 approaches in all the five case studies. Due to the complexity of some of the search environments, the

random-walk method often fails to find the source location within the allowed maximum search time, as evident from its poor success rate in Cases 1, 3, 4 and 5. Table 3 shows that *Bayes-Swarm* finds the primary source location about 5 to 100 times faster than exhaustive search in all five cases. As the random-walk reaches the goal only in the first two case studies, we compare *Bayes-Swarm* with the random-walk method only in these case studies; *Bayes-Swarm* is observed to perform 83 and 5 times faster than the random-walk method in case studies 1 and 2.

5.4 Experiment 4: Comparing Bayes-Swarm with a Swarm Intelligence Method

To perform further comparative analysis of Bayes-Swarm with a state-of-the-art method, the well-known Glowworm-based swarm search algorithm [19] is chosen. Specifically, we use the implementation of the Glowworm algorithm available at [38]. For this analysis, both algorithms are run on case study 5 (first problem in [35], further described in Appendix A.5). Both algorithms are run with the same robot specifications and environment settings as in [35]. It should be noted that there are two main differences between the generic mission objectives of the Bayes-Swarm algorithm and the Glowworm algorithm: 1) Bayes-Swarm is designed to find the source with maximum strength signal in the presence of other weaker (say decoy) sources, while the Glowworm algorithm is designed to find all local and global sources (both mission objectives can translate to important practical applications in the emergency response and defense domains). 2) The Glowworm algorithm assumes the robots to be initially distributed in the search arena, while Bayes-Swarm makes no such assumption. With regards to the first difference, we compare Bayes-Swarm’s completion time to find the global target source (with maximum signal strength) with the time that the Glowworm algorithm takes to find the first source (i.e., any source, local or global) – thus Bayes-swarm’s job is made to be at least as (or likely more) difficult. The second difference, w.r.t. starting points, is readily handled in our algorithm, since Bayes-Swarm is agnostic to the initial location of the robots.

We assume 50-robot teams and randomly generate the initial location of the robots in the arena, to be used by both methods: $-3 \leq x_1 \leq -1.2$ and $-3 \leq x_2 \leq 3$; based on the settings used in the reported Glowworm algorithm, the robot velocity is set at $V = 1\text{m/s}$ ². Since the Glowworm algorithm employs a stochastic search approach, it is run 10 times on this problem. It is observed that the robot team under *Bayes-Swarm* finds the source with maximum strength in 1.86 time units. In contrast, the robot team under the Glowworm algorithm takes 3.04 ± 0.4 time units (mean \pm std-dev. over 10 runs) to find the first (any) source, and 4.44 ± 0.55 time units

Table 3. Performance of the Bayes-Swarm, baseline, and competing algorithms on five test case studies with 5 robots; the exploitation coefficient of the Bayes-Swarm algorithm is set at 0.4 for all case studies, except the case study 5, which is set at 0.99.

Case Study	Algorithm	Total Time * [s]	Success Rate
1	<i>Bayes-Swarm</i>	246.1	1/1
	Random-Walk	20,394	1/5
	Exhaustive Search	22,174	1/1
2	<i>Bayes-Swarm</i>	42.5	1/1
	Random-Walk	227.6	5/5
	Exhaustive Search	225.3	1/1
3	<i>Bayes-Swarm</i>	260.1	1/1
	Random-Walk	-	0/5
	Exhaustive Search	22,174	1/1
4	<i>Bayes-Swarm</i>	373.2	1/1
	Random-Walk	-	0/5
	Exhaustive Search	9,163 [†]	1/1
5	<i>Bayes-Swarm</i>	31.9	1/1
	Random-Walk	-	0/5
	Exhaustive Search	992 [†]	1/1

* 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.

[†] 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).

to find the source with maximum signal strength. These results show that not only Bayes-Swarm is 58% faster than the Glowworm algorithm in finding the global target source, but also finds the global target source is less time than that taken by the Glowworm method to find any source.

It is important to note that the performance of both Bayes-Swarm and Glowworm algorithms are affected by their respective prescribed parameters. In the case of the Glowworm implementation, we used the same parameter settings as those recommended by Kaipa et al. [38] (i.e., the paper from which we adopt the implementation of this algorithm). In the case of Bayes-Swarm, it was readily evident from earlier parametric analysis (Fig. 6) that, for complex multimodal signal environments a value of $\alpha < 0.4$ works well. Hence, we explored how Bayes-Swarm would compare to the Glowworm algorithm when implemented with different values of $\alpha < 0.4$. The results show that Bayes-Swarm found the target with maximum strength in 2.43, 3.32, and 2.40 time units for α set at 0.05, 0.1, and 0.2, respectively; these mission completion times are still better than that resulting from the Glowworm implementation.

²These settings are purely computational, and are used here to preserve the sanctity of the Glowworm implementation, and allow fair comparison.

5.5 Discussion of Bayes-Swarm Performance

The various empirical analyses 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 α 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 α to the environment during the mission. Another important advantage of the Bayes-Swarm algorithm over model-free algorithms is its provisioning 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 downsampling is one of them. Since updating the Gaussian Process (GP) based belief model presents a cubic time complexity w.r.t. data-set size, we are required to downsample the collective data set of the swarm (observations made by agents). Without such downsampling, 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 downsampling approach based on sample rate compression. While this approach is simple to implement, it is far from optimal, especially for large swarm systems with 100's of robots. In such scenarios, choosing the most informative samples (to update the belief model) out of the entire set of observations remains a critical question, which 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 partial observance across the team due to communication range restrictions or communication intermittency issues. The allowance of asynchronous decision-making (currently offered by Bayes-Swarm) does help to some extent in mitigating the impact of such communication network limitations. However, in order to minimize potential conflict between agents' decisions under partial observance, further advancements are needed in the formulations of the acquisition function and constraints guiding the waypoint planning of swarm agents.

6 Conclusion

In this paper, we proposed an asynchronous and decentralized algorithm to guide the path planning of a team or swarm of robots that is searching for the source of a spatially

distributed signal in 2D arenas. This algorithm is founded on an extension of the batch Bayesian optimization method, with advancements made for application to embodied swarm systems. A new acquisition function is designed to be able to uniquely incorporate the following: 1) modeling knowledge gain over trajectories, as opposed to at points; 2) implicitly mitigating overlapping trajectories among robots to maximize unique knowledge gain; and 3) incentivising robots to reach (closest to) the expectation of the source, while accounting for constraints on the robot's motion and cost incurred by it in reaching a candidate waypoint. A heuristic (parameter, α) is currently used to balance the source seeking and knowledge gain components of the acquisition function, and thus further parametric analysis is performed to understand its impact. It is found that suitable values of this parameter depends both on the size of the swarm and the complexity of the signal's spatial distribution. An important direction of future research will be to build on this understanding to formulate a situation-adaptive variation (instead of user prescription) of the weighting coefficient.

To evaluate and compare the performance of the proposed algorithm, *Bayes-Swarm*, exhaustive search and random-walk baselines are considered. These algorithms are tested on five distinct case studies, with varying arena size and complexity (non-convexity) of the spatial distribution of the signal. Performance is analyzed in terms of completion time and mapping error. The *Bayes-Swarm* approach easily outperforms the exhaustive search and random-walk approaches by achieving up to 90 times better values of completion time. In addition, we compared our algorithm with the state-of-the-art SI-based Glowworm algorithm, over a benchmark (multi-source) problem, with the outcomes clearly demonstrating the search efficiency benefits of Bayes-Swarm.

Scalability of the *Bayes-Swarm* algorithm is also analyzed, with significant performance gain (in terms of super-linear reduction in completion time) observed as the swarm size is changed from 2 to 20, and then mostly saturating owing to the bounds on the size of the arena. Increased swarm size, while beneficial to the mission, also increases the rate at which signal data is collected; this then increases the online computational cost of updating the GP model of the signal environment by every robot during the mission. Thus future work will look at advanced downsampling-based update approaches (e.g., using active learning techniques) or direct sharing of model updates across robots (instead of sharing of data), especially for applications where 100's–1000's of robots are needed, or where longer mission time periods are needed. This, along with physical demonstration and the consideration of partial observability due to communication constraints, will allow us to more comprehensively explore the scalability of the *Bayes-Swarm* algorithm in the future.

Acknowledgements

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.

References

- [1] Baetz, W., Kroll, A., and Bonow, G., 2009. "Mobile robots with active ir-optical sensing for remote gas detection and source localization". In 2009 IEEE International Conference on Robotics and Automation, IEEE, pp. 2773–2778.
- [2] Nagatani, K., Kiribayashi, S., Okada, Y., Otake, K., Yoshida, K., Tadokoro, S., Nishimura, T., Yoshida, T., Koyanagi, E., Fukushima, M., et al., 2013. "Emergency response to the nuclear accident at the fukushima dai-ichi nuclear power plants using mobile rescue robots". *Journal of Field Robotics*, **30**(1), pp. 44–63.
- [3] Li, W., Farrell, J. A., Pang, S., and Arrieta, R. M., 2006. "Moth-inspired chemical plume tracing on an autonomous underwater vehicle". *IEEE Transactions on Robotics*, **22**(2), pp. 292–307.
- [4] Viseras, A., Wiedemann, T., Manss, C., Magel, L., Mueller, J., Shutin, D., and Merino, L., 2016. "Decentralized multi-agent exploration with online-learning of gaussian processes". In 2016 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 4222–4229.
- [5] Song, D., Kim, C.-Y., and Yi, J., 2012. "Simultaneous localization of multiple unknown and transient radio sources using a mobile robot". *IEEE Transactions on Robotics*, **28**(3), pp. 668–680.
- [6] De Silva, O., Mann, G. K., and Gosine, R. G., 2012. "Development of a relative localization scheme for ground-aerial multi-robot systems". In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, IEEE, pp. 870–875.
- [7] Ghassemi, P., and Chowdhury, S., 2018. "Decentralized task allocation in multi-robot systems via bipartite graph matching augmented with fuzzy clustering". In ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. V02AT03A014–V02AT03A014.
- [8] Ghassemi, P., DePauw, D., and Chowdhury, S., 2019. "Decentralized dynamic task allocation in swarm robotic systems for disaster response". In IEEE International Symposium on Multi-Robot and Multi-Agent Systems (MRS).
- [9] Pugh, J., and Martinoli, A., 2007. "Inspiring and modeling multi-robot search with particle swarm optimization". In Swarm Intelligence Symposium, 2007. SIS 2007. IEEE, IEEE, pp. 332–339.
- [10] Gunning, D., 2017. "Explainable artificial intelligence (xai)". *Defense Advanced Research Projects Agency (DARPA)*, nd Web.
- [11] Tan, Y., and Zheng, Z.-y., 2013. "Research advance in swarm robotics". *Defence Technology*, **9**(1), pp. 18–39.
- [12] Odonkor, P., Ball, Z., and Chowdhury, S., 2019. "Distributed operation of collaborating unmanned aerial vehicles for time-sensitive oil spill mapping". *Swarm and Evolutionary Computation*.
- [13] Sinha, A., Kaur, R., Kumar, R., and Bhondekar, A., 2017. "A cooperative control framework for odor source localization by multi-agent systems".
- [14] Wiedemann, T., Shutin, D., Hernandez, V., Schaffernicht, E., and Lilienthal, A. J., 2017. "Bayesian gas source localization and exploration with a multi-robot system using partial differential equation based modeling". In Olfaction and Electronic Nose (ISOEN), 2017 ISOCS/IEEE International Symposium on, IEEE, pp. 1–3.
- [15] Charrow, B., Michael, N., and Kumar, V., 2014. "Cooperative multi-robot estimation and control for radio source localization". *The International Journal of Robotics Research*, **33**(4), pp. 569–580.
- [16] Hajieghrary, H., Hsieh, M. A., and Schwartz, I. B., 2016. "Multi-agent search for source localization in a turbulent medium". *Physics Letters A*, **380**(20), pp. 1698–1705.
- [17] Sujit, P., and Ghose, D., 2009. "Negotiation schemes for multi-agent cooperative search". *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, **223**(6), pp. 791–813.
- [18] Kennedy, J., 2010. "Particle swarm optimization". *Encyclopedia of machine learning*, pp. 760–766.
- [19] Krishnanand, K., Amruth, P., Guruprasad, M., Bidargaddi, S. V., and Ghose, D., 2006. "Glowworm-inspired robot swarm for simultaneous taxis towards multiple radiation sources". In Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., IEEE, pp. 958–963.
- [20] Senanayake, M., Senthoooran, I., Barca, J. C., Chung, H., Kamruzzaman, J., and Murshed, M., 2016. "Search and tracking algorithms for swarms of robots: A survey". *Robotics and Autonomous Systems*, **75**, pp. 422–434.
- [21] Kennedy, J., 2006. "Swarm intelligence". In *Handbook of nature-inspired and innovative computing*. Springer, pp. 187–219.
- [22] Bonabeau, E., Marco, D. d. R. D. F., Dorigo, M., Théraulaz, G., Theraulaz, G., et al., 1999. *Swarm intelligence: from natural to artificial systems*. No. 1. Oxford university press.
- [23] Jatmiko, W., Sekiyama, K., and Fukuda, T., 2006. "A

- pso-based mobile sensor network for odor source localization in dynamic environment: Theory, simulation and measurement". In 2006 IEEE International Conference on Evolutionary Computation, IEEE, pp. 1036–1043.
- [24] Ghassemi, P., Lulekar, S. S., and Chowdhury, S., 2019. "Adaptive model refinement with batch bayesian sampling for optimization of bio-inspired flow tailoring". In AIAA Aviation 2019 Forum, p. 2983.
- [25] Kolling, A., Walker, P., Chakraborty, N., Sycara, K., and Lewis, M., 2016. "Human interaction with robot swarms: A survey". *IEEE Transactions on Human-Machine Systems*, **46**(1), pp. 9–26.
- [26] Ghassemi, P., and Chowdhury, S., 2019. "Decentralized informative path planning with balanced exploration-exploitation for swarm robotic search". In ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE, American Society of Mechanical Engineers.
- [27] Ghosh, S., 2001. "Understanding complex, real-world systems through asynchronous, distributed decision-making algorithms". *Journal of Systems and Software*, **58**(2), pp. 153–167.
- [28] Kandasamy, K., Krishnamurthy, A., Schneider, J., and Póczos, B., 2018. "Parallelised bayesian optimisation via thompson sampling". In International Conference on Artificial Intelligence and Statistics, pp. 133–142.
- [29] Rasmussen, C. E., 2003. "Gaussian processes in machine learning". In Summer School on Machine Learning, Springer, pp. 63–71.
- [30] Snoek, J., Larochelle, H., and Adams, R. P., 2012. "Practical bayesian optimization of machine learning algorithms". In Advances in neural information processing systems, pp. 2951–2959.
- [31] Morere, P., Marchant, R., and Ramos, F., 2017. "Sequential bayesian optimization as a pomdp for environment monitoring with uavs". In Robotics and Automation (ICRA), 2017 IEEE International Conference on, IEEE, pp. 6381–6388.
- [32] Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. W., 2012. "Information-theoretic regret bounds for gaussian process optimization in the bandit setting". *IEEE Transactions on Information Theory*, **58**(5), pp. 3250–3265.
- [33] Ginsbourger, D., Le Riche, R., and Carraro, L., 2007. "A multi-points criterion for deterministic parallel global optimization based on kriging". In NCP07.
- [34] Li, M., Lu, K., Zhu, H., Chen, M., Mao, S., and Prabhakaran, B., 2008. "Robot swarm communication networks: architectures, protocols, and applications". In 2008 Third International Conference on Communications and Networking in China, IEEE, pp. 162–166.
- [35] Kaipa, K. N., Puttappa, A., Hegde, G. M., Bidargaddi, S. V., and Ghose, D., 2006. "Rendezvous of glowworm-inspired robot swarms at multiple source locations: A sound source based real-robot implementation". In International Workshop on Ant Colony Optimization and Swarm Intelligence, Springer, pp. 259–269.
- [36] Wade, G., 1994. *Signal coding and processing*. Cambridge university press.
- [37] Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotocz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A., 2009. "The e-puck, a robot designed for education in engineering". In Proceedings of the 9th conference on autonomous robot systems and competitions, Vol. 1, Castelo Branco, Portugal, pp. 59–65.
- [38] Kaipa, K. N., and Ghose, D., 2017. *Glowworm swarm optimization: theory, algorithms, and applications*, Vol. 698. Springer.

A Definition of Case Studies

A.1 Case Study 1: Large arena, convex signal distribution

$$f = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_1\|^2}{130}\right) \quad (13)$$

Here, $\mathbf{x} = (x_1, x_2)$, where $0 \leq x_i \leq 24$, and $\mathbf{c}_1 = (5, 23)$. The initial feasible direction, $\Delta\theta$ is set at 90.

A.2 Case Study 2: Small arena, non-convex signal distribution

$$f = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_1\|^2}{3}\right) + \frac{1}{2} \exp(-2\|\mathbf{x} - \mathbf{c}_2\|^2) \quad (14)$$

Here, $\mathbf{x} = (x_1, x_2)$, where $0 \leq x_i \leq 2.4$, $\mathbf{c}_1 = (1.9, 2.3)$, and $\mathbf{c}_2 = (1.5, 0.5)$. The initial feasible direction, $\Delta\theta$ is set at 90.

A.3 Case Study 3: large arena, non-convex signal distribution

$$f = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_1\|^2}{120}\right) + \frac{1}{2} \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_2\|^2}{30}\right) \quad (15)$$

Here, $\mathbf{x} = (x_1, x_2)$, where $0 \leq x_i \leq 24$, $\mathbf{c}_1 = (10, 23)$, and $\mathbf{c}_2 = (15, 5)$. The initial feasible direction, $\Delta\theta$ is set at 90.

A.4 Case Study 4: Large arena, highly multi-modal signal distribution

$$f = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_1\|^2}{130}\right) + \frac{2}{5} \sum_{i=2}^7 \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{40}\right) \quad (16)$$

Here, $\mathbf{x} = (x_1, x_2)$, where $-24 \leq x_i \leq 24$. Moreover, $\mathbf{c}_1 = (21, 19)$, $\mathbf{c}_2 = (21, -19)$, $\mathbf{c}_3 = (0, -15)$, $\mathbf{c}_4 = (0, 15)$, $\mathbf{c}_5 = (-19, 10)$, $\mathbf{c}_6 = (21, 19)$, $\mathbf{c}_7 = (-15, -15)$. The initial feasible direction, $\Delta\theta$ is set at 360.

A.5 Case Study 5: Multi-modal source

We use the first example given in [35], which is a multi-modal function, and defined as follows:

$$f(x_1, x_2) = 3(1 - x_1)^2 \exp(-(x_1^2) - (x_2 + 1)^2) \quad (17)$$

$$- 10(x_1/5 - x_1^3 - x_2^5) \exp(-x_1^2 - x_2^2) \quad (18)$$

$$- (1/3) \exp(-(x_1 + 1)^2 - x_2^2) \quad (19)$$

Here, $-3 \leq x_i \leq 3$. A set of 50 robots are randomly deployed in a two-dimensional region such that to $-3 \leq x_1 \leq -1.2$ 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 $(-0.0093, 1.5814)$.

Table 5. The settings of the Glowworm algorithm (experiment 4).

N_r	ρ	γ	β	r_s	Δs [m]
50	0.4	0.6	0.08	3	0.03

N_r : Number of robots; ρ : Luciferin decay constant; γ : Luciferin enhancement constant; β : Decision range gain; r_s : Sensor range of robots; and Δs : Distance moved by each Glowworm when a decision is taken.

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.

Table 4. The settings of the Bayes-Swarm algorithm.

Experiment	Case Study	N_r	V [m/s]	T [s]	α
1	2	4	0.1	4	[0-1]
1	4	10	0.1	4	[0-1]
2	4	[2-100]	0.1	4	0.4
3	1-4	5	0.1	4	0.4
3	5	5	0.1	4	0.99
4	5	50	1	0.1	0.05, 0.1 0.2, 0.4

N_r : Number of robots; V : Velocity of robots; T : Decision-horizon length; and α : Exploitation coefficient.