Opportunistic Multi-robot Environmental Sampling via Decentralized Markov Decision Processes

Ayan Dutta[1], O. Patrick Kreidl[2], and Jason M. O'Kane[3]

[1] School of Computing, University of North Florida, USA,
a.dutta@unf.edu
[2] School of Engineering, University of North Florida, USA,
patrick.kreidl@unf.edu
[3] Department of Computer Science and Engineering,
University of South Carolina, USA,
jokane@cse.sc.edu

Abstract. We study the problem of information sampling with a group of mobile robots from an unknown environment. Each robot is given a unique region in the environment for the sampling task. The objective of the robots is to visit a subset of locations in the environment such that the collected information is maximized, and consequently, the underlying information model matches as close to reality as possible. The robots have limited communication ranges, and therefore can only communicate when nearby one another. The robots operate in a stochastic environment and their control uncertainty is handled using factored Decentralized Markov Decision Processes (Dec-MDP). When two or more robots communicate, they share their past noisy observations and use a Gaussian mixture model to update their local information models. This in turn helps them to obtain a better Dec-MDP policy. Simulation results show that our proposed strategy is able to predict the information model closer to the ground truth version than compared to other algorithms. Furthermore, the reduction in the overall uncertainty is more than comparable algorithms.

Keywords: Information Sampling, Markov Decision Process, Gaussian Mixture

1 Introduction

Coverage path planning by a group of autonomous mobile robots has many real-world applications including area cleaning, painting, and precision agriculture and the problem has been extensively studied in the literature. The goal in this task is to cover all the locations in the environment [7]. Recently, researchers have looked into more constrained scenarios, in which the robots can only visit a subset of points in the environment due to the budget constraints, while collecting maximal information from an unknown environment [5, 10, 12, 14, 19].

We study such a multi-robot information sampling problem in this paper. This problem is known to be NP-hard problem to solve optimally [19].

In a real-world setting, the robots not only have a budget constraint, but they also have limited communication ranges, and therefore, they are not always guaranteed to maintain a global communication network unless the underlying control mechanism makes them do so continuously or periodically [1]. We use a less restrictive model in which connectivity is *opportunistic* — the robots form ad-hoc local networks with the nearby robots whenever possible. The robots *locally* share their history of noisy sensor measurements for better developing the global information model. This significantly reduces the communication overhead and execution time compared to continuous connectivity models [5,12]. On the other hand, for applications in extreme environmental conditions, e.g., ocean surface mapping [3], the control of the robots becomes stochastic. Our presented solution gracefully handles this uncertainty by modeling the planning problem as a Decentralized Markov Decision Process (Dec-MDP), where the coordinating robots share a joint reward system while their state and action spaces are independent. Simulation results show that our proposed approach is up to 71.68% faster than a comparable continuous connectivity approach while performing at par in terms of the modeling of the underlying information field.

Our primary contributions in this paper are two-fold:

- First, to the best of our knowledge, this is the first work that employs a decentralized MDP technique for multi-robot information collection under control uncertainty.
- Secondly, we address another practical challenge, i.e., limited communication ranges of the robots, by developing an opportunistic connectivity-based novel decentralized coordination mechanism.

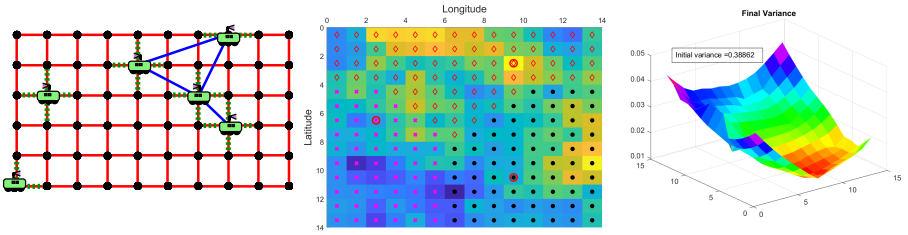
2 Related Work

Autonomous mobile robots are used for information collection in real-world applications such as precision agriculture, search and rescue, monitoring, among others. One of the first approaches is due to Krause et al. [10], who proposed greedy strategies to find the informative locations to place a set of sensors, utilizing Gaussian Processes to model the phenomena [16]. Singh et al. [19] proposed the first informative path planning solution for mobile robots. A decentralized multi-robot online informative sampling method is proposed by Viseras et al. [20]. Similar to ours, the sensing is assumed to be noisy. However, the robots exchange a significant amount of information (e.g., past visited locations and corresponding measurements, next locations, etc.), which might be infeasible to achieve in a real-world setting. Similar to our work, Luo and Sycara partition the environment *a priori* and assign each robot to a unique Voronoi cell. Region partitioning has also been used for multi-robot information collection in [4,6,9]. A multi-robot information collection approach with dynamic goal location planning is proposed in [14]. Most of these studies introduce centralized methods, which do not take robots' communication constraints into account. In the real

world, the robots have limited communication ranges, which poses a challenge for coordinated environmental sampling. A survey of various connectivity strategies is presented in [1]. Three primary connectivity methods are found: periodic [17], continuous [5], and no requirement, e.g., opportunistic connectivity [4]. The first two requirements are more stringent — the robots have to plan their future locations jointly. However, in an opportunistic setting, the robots' primary goal is to collect maximal information. When two or more robots come within each other's communication ranges, they share their findings in order to make more informative decision in the future [4]. Although real robots exhibit stochastic motion in applications such as underwater monitoring, most of the prior work on multi-robot information collection does not handle control uncertainty. The only informative sampling work for a single robot, to the best of our knowledge, that models stochastic motion using a Markov Decision Process (MDP) is due to Ma et al. [13]. In this work, we consider n robots instead of one, and thus we propose a decentralized (Dec) MDP-based coordination technique for information collection under control uncertainty. An optimal solution for Dec-MDP is proposed in [2], but it is not scalable to large multi-robot systems due to its NEXP-completeness. A heuristic solution is presented in [15]. Our Dec-MDP solution is greedy for better scalability, similar to the ones proposed in [11, 18].

3 Problem Setup and Basic Algorithm

A homogeneous team of n mobile robots $r_1, r_2, \ldots, r_n$ moves through a shared planar environment. Each robot is equipped with sufficient on-board sensing (i.e. GPS) to localize itself within the environment, a sensor that measures a phenomenon of interest at the robot's current location, and a communication device that enables limited-range communication with other nearby robots. Let $\mathcal{V}$ denote a given finite set of information collection points, or *nodes*, that cover the environment in a grid pattern. Each robot r_i, starting from a unique node s_0^i in $\mathcal{V}$, is responsible for a subset of nodes $\mathcal{V}_i$ containing s_0^i such that $\mathcal{V} = \cup_{i=1}^{n} \mathcal{V}_i$ and, for every other robot r_j, the subsets $\mathcal{V}_i$ and $\mathcal{V}_j$ are disjoint. We use k-medoids clustering to achieve such a partitioning [8]. The centroids of the partitions are selected to be the start nodes. (Other partitioning techniques such as Voronoi partitioning or k-means can also be used without affecting our presented solution.) The robots, having common knowledge of this size-n partition, each move sequentially over time in the cardinal directions to adjacent nodes within their own regions, until a given movement budget B expires. The outcome of each action in finite set $\mathcal{U}$ is stochastic, modeled by a transition probability function $f : \mathcal{V} \times \mathcal{U} \times \mathcal{V} \to [0, 1]$, under which $f(s, u, s')$ represents the probability for arriving at node s' upon executing action u at node s. For example, this transition probability function f should assign high probability to cardinal movements in the intended direction, and smaller probabilities to movements that represent imperfect movements. The robots are interested in some ambient real-valued phenomenon, which varies across the environment. We model this phenomenon as a random vector $\mathbf{X}$, so that component X_s denotes the value of the phe-



(a) (b) (c)

Fig. 1. (a) Illustration of the problem setup with six robots moving on a grid of nodes from which to gather information. Nearby robots may communicate either directly or via multiple hops. (b) A specific 14-by-14 node grid (with the spatially-varying "ground truth" phenomenon conveyed by the blue/yellow shading) to be explored by three robots with given start nodes (red circles) and component static partitions (color-coded node markers). (c) An instance of the initial and final per-node variances, quantifying the reduction in prediction uncertainty, using our proposed solution ($n = 3$).

nomenon at node s. *The multi-robot objective is to navigate the environment in order to maximally reduce prediction uncertainty in* $\mathbf{X}$ *subject to the budget constraint.* Figure 1 illustrates (a) the conceptual setup as well as (b) a specific partition instance of the problem and (c) the character of its solution in our simulation experiments. The following subsections detail the basic models and algorithms by which the robots pursue this objective *in the absence of inter-robot communication*, leaving the extensions to leverage communication for Section 4.

3.1 Prediction via Gaussian Processes (GPs)

We use the Gaussian Process (GP) to model the uncertain environment, specifically assuming that (i) the phenomenon of interest at every node takes a scalar real value and (ii) all nodes generate information according to a length-$|\mathcal{V}|$ Gaussian random vector $\mathbf{X}$ with known (prior) mean vector μ and covariance matrix Σ. It is well known that the optimal prediction, in the sense of minimum mean square error, is the mean vector μ for which the covariance matrix Σ characterizes the prediction's uncertainty [16]. Its (differential) entropy, a volumetric measure of that uncertainty, is given by $H(\mathbf{X}) = \frac{1}{2}\log|\Sigma| + \frac{|\mathcal{V}|}{2}\log(2\pi e)$, where $|\mathcal{V}|$ denotes set cardinality but $|\Sigma|$ denotes matrix determinant.

Each robot's sensing process is imperfect, specifically assuming that measurements are corrupted by zero-mean stationary additive white Gaussian noise, independently and identically distributed (across robots and nodes) with variance σ_n^2. We suppose that (i) all robots are initialized with the same prior model $GP^0 = \{\mu, \Sigma\}$ and (ii) each robot r_i takes measurement y_0^i at its start node s_0^i. It follows that, before any movement decisions are made, the model GP_0^i local to robot r_i is given by (posterior) statistics

$$\begin{aligned} \Sigma_0^i &= \Sigma - \Sigma\mathbf{C}\left(s_0^i\right)' \left(\mathbf{C}\left(s_0^i\right)\Sigma\mathbf{C}\left(s_0^i\right)' + \sigma_n^2\mathbf{I}\left(s_0^i\right)\right)^{-1} \mathbf{C}\left(s_0^i\right)\Sigma \\ \mu_0^i &= \mu + \Sigma_0^i\mathbf{C}\left(s_0^i\right)' \left(y_0^i - \mathbf{C}\left(s_0^i\right)\mu\right)/\sigma_n^2 \end{aligned} \tag{1}$$

where $\mathbf{C}\left(s_0^i\right)$ denotes the length-$|\mathcal{V}|$ row vector of all zeros except for a one in component s_0^i, $\mathbf{C}\left(s_0^i\right)'$ is its matrix transpose (i.e., the analogous column vector) and $\mathbf{I}\left(s_0^i\right)$ denotes the 1-by-1 identity matrix (i.e., the scalar value of one).

The matrix notation within (1) prepares for processing measurements in batch. That is, suppose $\mathbf{y}$ denotes a length-p column vector representing a sequence of collected measurements and let $\mathbf{s}$ be the associated sequence of visited nodes (possibly with repetition). Then, the posterior second-order statistics $\mu_{\mathbf{X}|\mathbf{Y}}$ and $\Sigma_{\mathbf{X}|\mathbf{Y}}$ are computed via (1) upon letting $\mathbf{C}(\mathbf{s})$ denote the p-by-$|\mathcal{V}|$ matrix whose rows consist of successive unit vectors, each having direction corresponding to the successive components of $\mathbf{s}$, while $\mathbf{I}(\mathbf{s})$ denotes the p-by-p identity matrix. Under our noise assumptions, whether (1) is implemented in batch or recursively over measurements gives the same statistics: the resulting mean $\mu_{\mathbf{X}|\mathbf{Y}}$ is the minimum mean-square-error predictor of $\mathbf{X}$, given $\mathbf{Y} = \mathbf{y}$, and the resulting covariance $\Sigma_{\mathbf{X}|\mathbf{Y}}$ analogously implies posterior entropy $H(\mathbf{X}|\mathbf{Y})$.

A final remark concerns the determinant $|\Sigma|$ being an expensive computation, scaling roughly cubically with matrix dimensions. It is common (e.g., in kernel-based parametrizations of GPs) that there are diminishing correlations among nodes as the distance between them grows. This motivates approximation of its determinant by the product of the per-node variances $\sigma_1^2, \sigma_2^2, \sigma_3^2, \ldots$ along its diagonal e.g., $\log|\Sigma| \approx \sum_{s=1}^{|\mathcal{V}|} \log(\sigma_s^2)$. The approximation is, in fact, an upper bound on the true determinant (via Hadamard's inequality), achieving equality if and only if the matrix is truly diagonal. It follows for posterior entropy that

$$H(\mathbf{X}|\mathbf{Y}) \leq \sum_{s=1}^{|\mathcal{V}|} H(X_s|\mathbf{Y}) \quad \text{with} \quad H(X_s|\mathbf{Y}) = \frac{1}{2}\log\left(2\pi e \sigma^2_{s|\mathbf{Y}}\right) \quad . \tag{2}$$

We utilize these per-node entropies $\{H(X_s|\mathbf{Y}); s \in \mathcal{V}\}$ to define a reward function that directs the robots' movements, as described in the next section.

3.2 Control via Markov Decision Processes (MDPs)

Before deployment, the robots are provided with a common set of initial training data $\mathcal{D}$ to generate their local initial GP models, GP^i, and calculate the local hyper-parameters. The initial rewards are then calculated based on GP^i using (2). In a single-robot system, to handle the control uncertainty, we can model the problem as a Markov Decision Process (MDP), where the states are represented by the node set $\mathcal{V}$ and the reward can be calculated using (2). We can use standard algorithms such as value iteration or policy iteration to solve the MDP and compute an optimal policy π, which the robot can then follow to collect information. At the start of the multi-robot deployment, similar to a single-robot system, each robot r_i generates such an initial policy π^i by which control $u_0^i \in \mathcal{U}$ is selected and the first (stochastic) move from node s_0^i to node s_1^i is realized. We assume temporarily (for the simplified presentation in this section) that *no communication is available*, so multi-robot system reduces to n independent single-robot systems, each having its local π^i and GP^i. When a

new node is visited by robot r_i, it senses the data at the node, incorporates the new observation into its local GP^i and estimates the posterior statistics given by (1), and finally executes the next control, following the policy π^i. This *sense-estimate-move* cycle repeats until the budget of r_i is exhausted. As this process proceeds, r_i will adapt its policy π^i, by re-planning using its updated local model GP^i, after every τ cycles to reflect its improved knowledge of the phenomenon.

4 Algorithm with Opportunistic Communication

Section 3 characterized the basic algorithms by which each robot r_i explores its own subset of nodes $\mathcal{V}_i$ in the absence of inter-robot communication. We certainly expect improved prediction and control when *full* inter-robot communication during navigation is permitted, but we consider the challenge of a connectivity model that is only *opportunistic*, meaning we neither require the robots to remain in continuous communication with each other nor to plan to establish communication after a certain time interval. Instead, the robots focus primarily upon exploring in essentially the same distributed fashion described in Section 3, communicating during exploration only when within each other's communication ranges. Let CR denote the maximum communication range and define $R_k^i = \{r_j \mid ||s_k^i - s_k^j|| \leq CR\}$ as the set of robots that are within r_i's communication range in stage k, where s_k^i denotes the location of r_i at stage k. Thus $\bar{R}_k^i = R_k^i \cup \{r_i\}$ forms a connected communication graph such that any two robots in this network can send/receive messages from each other either directly or via hops. Note that, because communication is assumed to be symmetric, all the robots in $\bar{R}_k^i$ form the same local communication graph.

4.1 Prediction via GP Mixtures

With respect to prediction, the main challenge raised by distributed control with only opportunistic communication is that robots, because each explores a unique subset of nodes, are likely to meet possessing GPs that imply different posterior statistics. We thus seek a method by which to combine, or "fuse," different Gaussian statistics. The approach taken in [12], albeit under continuous connectivity, casts the problem as learning parameters within a set of Gaussian mixtures from data via an Expectation-Maximization (EM) algorithm. We shall adopt the same approach, but suitably modified for only opportunistic communication.

Recall from Section 3 that, in the basic algorithm, our use of GPs results in two essential outputs: the mean vector that represents the minimum-mean-square-error prediction of process $\mathbf{X}$, and the per-node variances that determine the per-node entropies $H(X_s)$. Consider assigning a length-n probability vector $(q_s^1, q_s^2, \ldots, q_s^n)$ to every node s in $\mathcal{V}$, where q_s^i represents the probability that component X_s is described by the Gaussian statistics of robot r_i. Denoting for each i the associated mean and variance by m_s^i and v_s^i, respectively, the Gaussian

mixture's statistics describing X_s are given by

$$m_s^* = \sum_{i=1}^{n} q_s^i m_s^i \quad \text{and} \quad v_s^* = \sum_{i=1}^{n} q_s^i \left(v_s^i + (m_s^i)^2 - (m_s^*)^2\right) \quad . \tag{3}$$

The statistics in (3) may be used to obtain the same essential outputs discussed in the basic algorithm: the mean vector $\mathbf{m}^* = (m_1^*, m_2^*, \ldots, m_{|\mathcal{V}|}^*)$ represents the fused minimum-mean-square-error prediction of process $\mathbf{X}$, while the per-node variances v_s^* permit approximate per-node entropies via $H(X_s) \approx \frac{1}{2}\log(2\pi e v_s^*)$.

We proceed to describe the EM algorithm that determines (i) the per-node mixture probabilities $\{q_s^i\}$ as well as (ii) the associated n per-robot Gaussian statistics i.e., per-node means $\{m_s^i\}$ and variances $\{v_s^i\}$ of each robot r_i The data in the algorithm is a batch of p measurements, in the sense discussed for (1) in Section 3 i.e., length-p column vectors $\mathbf{y}$ and $\mathbf{s}$ that represent, respectively, the collected measurements (possibly by multiple robots) and the associated sequence of visited nodes (possibly with repetition). The algorithm is iterative, initialized assuming n distinct GPs are available, each associated with a mean vector μ^i and covariance matrix Σ^i representing the prior statistics local to robot r_i. For every node s in $\mathcal{V}$ and index $i = 1, 2, \ldots n$, assign

$$m_s^i := \mathbf{C}(s)\mu^i, \quad v_s^i := \mathbf{C}(s)\Sigma^i\mathbf{C}(s)' \quad \text{and} \quad q_s^i := \begin{cases} 1-\epsilon & \text{, if } s \in \mathcal{V}_i \\ \epsilon/(n-1) & \text{, otherwise} \end{cases}$$

with matrix $\mathbf{C}$ as defined in (1) and $0 < \epsilon \ll 1$ denoting a "small" probability. The algorithm then repeats the following two-step procedure until convergence:

Expectation: For every node s in $\mathcal{V}$, denote by $\mathbf{y}_s$ the subvector of batch measurements $\mathbf{y}$ collected at node s and, for every index $i = 1, 2, \ldots n$, assign q_s^i proportional to $1/v_s^i$ if $\mathbf{y}_s$ is empty and otherwise assign q_s^i proportional to $q_s^i L_s^i$, where $L_s^i(\mathbf{y}_s)$ denotes the likelihood of $\mathbf{y}_s$ assuming independent measurements under a (univariate) Gaussian PDF with mean m_s^i and variance v_s^i.

Maximization: For every index $i = 1, 2, \ldots n$, denote by $\mathbf{y}^i$ the length-p^i subvector of batch measurements collected by robot r_i and by $\mathbf{s}^i$ the associated sequence of visited nodes. If $\mathbf{y}^i$ is empty, assign $\Lambda^i := \Sigma^i$ and $\nu^i := \mu^i$; otherwise,

$$\begin{aligned} \Lambda^i &:= \Sigma^i - \Sigma^i\mathbf{C}\left(\mathbf{s}^i\right)'\left(\mathbf{C}\left(\mathbf{s}^i\right)\Sigma^i\mathbf{C}\left(\mathbf{s}^i\right)' + \sigma_n^2\mathbf{Q}\left(\mathbf{s}^i\right)^{-1}\right)^{-1}\mathbf{C}\left(\mathbf{s}^i\right)\Sigma^i \\ \nu^i &:= \mu^i + \Lambda^i\mathbf{C}\left(\mathbf{s}^i\right)'\mathbf{Q}\left(\mathbf{s}^i\right)\left(\mathbf{y}^i - \mathbf{C}\left(\mathbf{s}^i\right)\mu^i\right)/\sigma_n^2 \end{aligned} \tag{4}$$

with $\mathbf{Q}\left(\mathbf{s}^i\right)$ denoting the p^i-by-p^i diagonal matrix of probabilities q_s^i ordered along the diagonal in correspondence with subvector $\mathbf{s}^i$. Then, for every node s in $\mathcal{V}$ and index $i = 1, 2, \ldots n$ assign $m_s^i := \mathbf{C}(s)\nu^i$ and $v_s^i := \mathbf{C}(s)\Lambda^i\mathbf{C}(s)'$.

The EM initialization assumes all robots' prior models $\{GP_{k-1}^i\}$ are available, which is not necessarily the case under only opportunistic communication. For example, consider the stage-0 perspective of a particular robot r_i, who will know only its local model GP_0^i and the prior model GP^0 from which it evolved. It can simply assign $GP_0^j := GP^0$ for every $j \neq i$ and proceed in subsequent stages as described. Of course, until robot r_i enters a stage k in which communication with

another robot r_j is possible (i.e., until R_k^i is non-empty), the non-local models will not change (i.e., the maximization step renders $GP_k^j := GP_{k-1}^j$) and thus be mismatched from the stage-k prior model local to robot r_j. Stages k with R_k^i non-empty begin with synchronizing the prior models $\{GP_{k-1}^i \mid r_i \in \bar{R}_k^i\}$ and sharing the local measurements $\{(\mathbf{y}^j, \mathbf{s}^j) \mid r_j \in \bar{R}_k^i\}$ to form batch data $(\mathbf{y}, \mathbf{s})$ upon which each connected robot proceeds to locally execute its EM algorithm. It should be noted that the algorithm concludes with synchronised posterior models $\{GP_k^j \mid r_j \in \bar{R}_k^i\}$, but if any models $\{GP_k^j \mid r_j \notin \bar{R}_k^i\}$ over disconnected robots remain mismatched then the statistics in (3) will also likely differ.

4.2 Control via Decentralized MDPs

For multi-robot systems, where one robot's reward might be affected by the observations made by the other robots, we can extend the n independent MDP model to Decentralized MDPs (Dec-MDP). We consider a factored-state (the local states are unique to the robots, e.g., the node set is partitioned into n subsets), transition-independent, and non-reward-independent Dec-MDP model [2,11,15]. This is due to the fact that the robots are placed in unique regions in the environment, but following (2), one robot's local reward is affected by other robots' sensed information. Unfortunately, this has been proved to be a NP-complete problem to solve optimally [2], and therefore, we adopt a greedy strategy to solve it [18]. The pseudocode for the approach is presented in Algorithm 1. Essentially, each robot $r_j \in R_k^i \cup r_i$ updates its GP model following the formulation in Section 4.1. Next, r_j augments its local MDP with the updated joint rewards calculated using (2) and generates a new local optimal policy to follow.

5 Evaluation

We have tested our proposed opportunistic online information sampling planner in simulation using MATLAB. The experiments are run on a laptop computer with a 1.80 GHz. Intel Core i7-8500U Processor, 16 GB RAM. We varied the number of robots between $\{2, 4, 6\}$. The robots were placed in a 4-connected grid environment of size 14×14 meters having unit-length square cells. Each robot was given a budget of 20 meters of travel. The policy update frequency, τ, is set to 5, a fraction of B. We use the Value Iteration algorithm available in the MDP-toolbox (https://bit.ly/38PPPcf) to obtain the policies. Our ground truth environment is modeled by a zero-mean Gaussian random vector $\mathbf{X} = (X_1, X_2, \ldots, X_{196})$ with covariance matrix built upon an exponential kernel function: specifically, for any pair of nodes s and t at spatial locations $\mathbf{p}_s$ and $\mathbf{p}_t$, respectively, let $\mathrm{Cov}(X_s, X_t) = \beta^2 \exp(-||\mathbf{p}_s - \mathbf{p}_t||/\ell)$, where hyperparameters $\beta > 0$ is the local standard deviation and ℓ (in meters) is the exponential rate of diminishing covariance between increasingly distant states. Our experiments assume $\beta = 1$ and $\ell = 25$ and then sample the resulting isotropic Gaussian Markov random field to simulate ground truth; Figure 1(b) depicts such an instance. The noisy sensing process at any node is simulated by adding to its ground truth value

Algorithm 1: Decentralized Environmental Sampling Using Opportunistic Connectivity and Dec-MDP

Procedure `sampleInformation()`
  Input: $\mathcal{V}_i \leftarrow$ robot r_i's unique, static partition – calculated offline.
  $B \leftarrow$ Exploration budget of the robots.
  $k \leftarrow$ the number of total moves, initially set to 0.
  Each robot r_i will follow the exploration scheme – <Move, Sense, Connect, Estimate, Adapt> – within $\mathcal{V}_i$:
  Sense data in the starting node and add it to the initial training set.
  Each robot 1) begins having the same prior GP learned from the initial training set, 2) then updates to GP_0^i using the measurement at the start node and 3) predicts the initial per-node entropies.
  $\pi_0^i \leftarrow r_i$'s initial MDP policy based on initial rewards.
  while $k < B$ **do**
    $k \leftarrow k + 1$.
    $R_k^i \leftarrow \emptyset$.
    Move to the next node s_k^i following the local policy π_{k-1}^i.
    Sense data in the current node; add the observation to the training set.
    Broadcast message and update R_k^i (Section 4.1).
    if $R_k^i \neq \emptyset$ **then**
      if r_i *has previously encountered with some robots* $R'' \subseteq R_k^i$ **then**
        Use the mixture parameters from this last encounter along with the newly observed data to update the local Gaussian model GP_k^i.
      else
        Share all observed data with $r_j \in R_k^i$ and use the EM algorithm to update GP_k^i.
      Update the rewards (2) based on GP_k^i.
      Adapt by executing *solveDecMDP*(R_k^i) and updating the local MDP policy $\pi_k^i \leftarrow \pi_i^*$.
    else
      Estimate Use the newly locally observed data and update GP_k^i.
    Adapt the local policy π_k^i based on revised rewards from GP_k^i after every τ cycles.

Procedure `solveDecMDP()`
  Input: $R' \leftarrow$ robots that are within CR distance of robot r_i
  Output: Solution of the local MDP of robot r_i
  $MDP_i^* \leftarrow$ Augmented local MDP with the joint reward function for $\forall r_j \in R'$.
  $\pi_i^* \leftarrow$ Solve MDP_i^* using the Value/Policy Iteration algorithm.
  return π_i^*.

a sample from the zero-mean univariate Gaussian distribution with variance $\sigma_n^2 = 0.25$. The probability parameters in the EM algorithm are set to $\epsilon = 10^{-3}$ and $\delta = 10^{-4}$.

We measured the performance of our approach by testing five other variants: 1) control strategies were varied between Dec-MDP and greedy — in the Dec-MDP variant, robots' actions are decided based on the MDP policy and in the greedy variant, a greedy action, i.e., that maximizes the one-step reward is chosen; 2) connectivity strategies were varied between no communication (NC), opportunistic (OC), and continuous communication (CC) among the robots. In cases of CC and NC, the robots were assumed to have infinite and zero communication ranges respectively. Note that the greedy-CC strategy is similar to the centralized technique proposed in [12]. In case of OC, the CR is set to 0.3 times of the environment's diagonal. Ten trials were conducted for each scenario.
Results. First, we are interested in investigating the most important metric to measure the performance of the proposed multi-robot information sampling approach – Mean Square Error (MSE), which depicts how closely the robots could model the underlying information field. The average MSE between the final predicted measurements and the ground truth measurements for different robot counts are shown in Figure 2. The standard deviation of our algorithm's yielded MSE is also shown in shaded blue. As can be observed, for $n \in \{2, 4, 6\}$, average MSE over time has reduced. Although this is true for all the implemented algorithms, for our proposed one (Dec-MDP-OC), the final MSE is one of the lowest amongst all. As expected, if there is no communication available, regardless of the control strategies, the MSE values are the highest, indicating a worse difference between the predicted and the ground truth information model. Similarly, if there is continuous communication, regardless of the control strategy, the average MSE is always better (lower) than our approach. However, the maximum final difference is only 11% with $n = 4$. Recall that this approach requires continuous communication among the robots, which is not only non-trivial to maintain [5], but also incurs high run-time cost. Our algorithm outperforms the comparable greedy strategy except for $n = 4$, since our approach looks into an infinite horizon to find the solution as opposed to the one-step look ahead used by the greedy strategy.

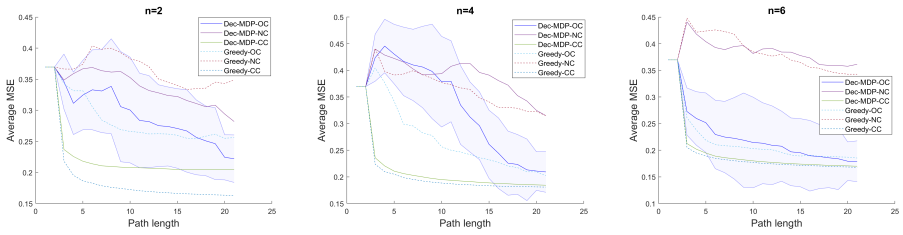


Fig. 2. Comparison of the MSE metric among the algorithms [lower is better].

A similar trend can be seen for the variance metric as well. The results for this metric are presented in Fig. 3. We have calculated the per-node variance after every GP prediction and the average across n robots are shown here. As discussed

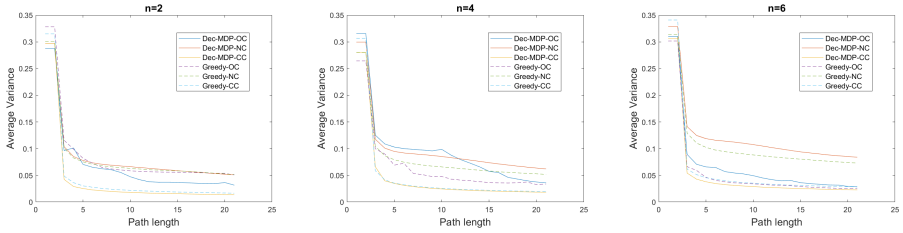
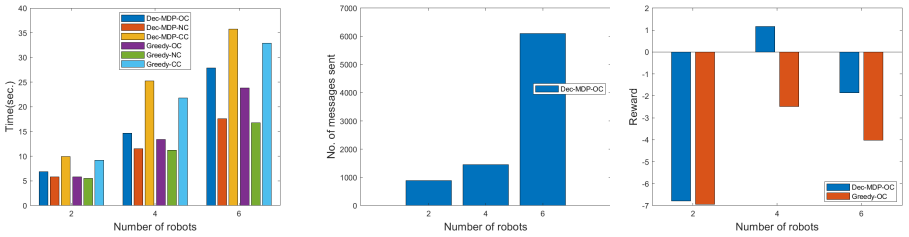


Fig. 3. Comparison of the variance metric among the algorithms [lower is better].

earlier, the variance indicates the uncertainty in the Gaussian Process prediction. Similar to MSE, the average variance decreases as the robots visit more nodes in the environment. In case of NC, the variance does not decrease as sharply as in the OC and CC cases since there is no Gaussian mixture process involved. On the other hand, the proposed Dec-MDP-OC strategy performs better or at par with the greedy-OC method while averaging very close to the CC strategies (the minimum being only 50% higher than the Dec-MDP-CC solution with $n = 6$).

Next, we investigated the run times of the proposed solution and compare it against other implemented algorithms. As can be seen in Fig. 4.(a), the run times of the proposed Dec-MDP-OC strategy are low and grow in a quadratic fashion with n. For example, with $n = 6$, the run time for the proposed approach is only 27.87 sec., whereas for the Dec-MDP and greedy-CC approaches they are 35.78 and 32.89 sec. respectively. A *connected component* is defined as a maximal set of robots for which the member robots can communicate directly or via multiple hops. These are the robots that participate in the proposed Gaussian mixture-based coordination strategy. The lower-bound on the number of connected components in a communication graph is 1, which essentially represents a CC strategy at that time. However, we have found that in the presented OC strategy, the connected component count is always greater than 1, which indicates a lower communication and coordination overhead than the centralized CC strategy such as used in [12]. This is also supported by the fact that the greedy and Dec-MDP CC strategies take more time than our OC technique. The number of messages required to be sent for the Gaussian mixture algorithm to be converged is also negligible, as shown in Fig. 4.(b).

The average reward collected by the robots is reported in Fig. 4.(c). This result shows that our proposed Dec-MDP-based control mechanism always collects higher reward than the greedy approach with the greatest difference occurring with $n = 4$. Finally, we visually compare the ground truth information field against the final predicted model across various algorithms. We can notice in Figs. 5 that the proposed Dec-MDP-OC approach makes more fine-grained and close-to-reality prediction than that of the greedy-OC and the NC strategies. This observation is supported by the numerical MSE data presented in Fig. 2.



(a) (b) (c)

Fig. 4. Comparison of a) Run times, b) Average number of messages sent by each robot, and c) Collected average rewards.

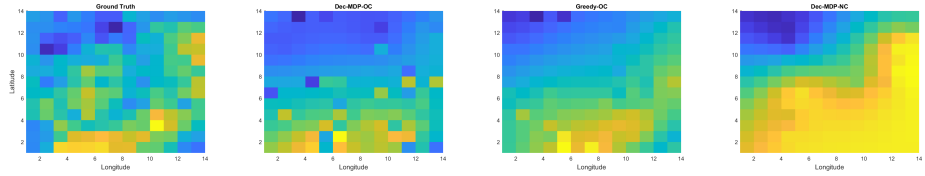


Fig. 5. Solution instance: visual comparison of predicted models by various algorithms against the ground truth measurements ($n = 2$).

6 Conclusions and Future Work

We have proposed an online multi-robot information sampling technique for an unknown environment. Our presented approach gracefully handles real-world constraints such as limited communication ranges and stochastic motion of the robots. The proposed strategy relies on robots' opportunistic communication patterns instead of forcing the robots to stay in continuous communication or connect after regular intervals. Results show that this technique can reduce the robots' run times significantly while performing at par in terms of modeling the underlying information field and reducing the uncertainty in the environment, compared to (potentially centralized) techniques that maintain continuous communication connectivity. In the future, we plan to investigate a better mixture method, in which each robot will maintain a history of past coordinated Gaussian mixtures. Finally, we plan to test the proposed solution with real robots for applications in domains such as precision agriculture and underwater robotics.

Acknowledgements. A. Dutta and O.P. Kreidl are partially supported by NSF CPS grant #1932300.

References

1. Amigoni, F., Banfi, J., Basilico, N.: Multirobot exploration of communication-restricted environments: A survey. IEEE Intelligent Systems **32**(6), 48–57 (2017)

2. Becker, R., Zilberstein, S., Lesser, V., Goldman, C.V.: Solving transition independent decentralized markov decision processes. JAIR **22**, 423–455 (2004)
3. Delight, M., Ramakrishnan, S., Zambrano, T., MacCready, T.: Developing robotic swarms for ocean surface mapping. In: ICRA (2016)
4. Dutta, A., Bhattacharya, A., Kreidl, O.P., Ghosh, A., Dasgupta, P.: Multi-robot informative path planning in unknown environments through continuous region partitioning. International Journal of Advanced Robotic Systems **17**(6), 1–18 (2020)
5. Dutta, A., Ghosh, A., Kreidl, O.P.: Multi-robot informative path planning with continuous connectivity constraints. In: ICRA (2019)
6. Fung, N., III, J.G.R., Nieto, C., Christensen, H.I., Kemna, S., Sukhatme, G.S.: Coordinating multi-robot systems through environment partitioning for adaptive informative sampling. In: ICRA, pp. 3231–3237. IEEE (2019)
7. Galceran, E., Carreras, M.: A survey on coverage path planning for robotics. Robotics and Autonomous systems **61**(12), 1258–1276 (2013)
8. Kaufmann, L.: Clustering by means of medoids. In: Proc. Statistical Data Analysis Based on the L1 Norm Conference, Neuchatel, 1987, pp. 405–416 (1987)
9. Kemna, S., Rogers, J.G., Nieto-Granda, C., Young, S., Sukhatme, G.S.: Multi-robot coordination through dynamic voronoi partitioning for informative adaptive sampling in communication-constrained environments. In: ICRA, pp. 2124–2130. IEEE (2017)
10. Krause, A., Singh, A., Guestrin, C.: Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. JMLR **9**(Feb), 235–284 (2008)
11. Kumar, R.R., Varakantham, P., Kumar, A.: Decentralized planning in stochastic environments with submodular rewards. In: AAAI, pp. 3021–3028 (2017)
12. Luo, W., Sycara, K.P.: Adaptive sampling and online learning in multi-robot sensor coverage with mixture of gaussian processes. In: ICRA, pp. 6359–6364. IEEE (2018)
13. Ma, K., Liu, L., Sukhatme, G.S.: An information-driven and disturbance-aware planning method for long-term ocean monitoring. In: IROS, pp. 2102–2108. IEEE (2016)
14. Ma, K.C., Ma, Z., Liu, L., Sukhatme, G.S.: Multi-robot informative and adaptive planning for persistent environmental monitoring. In: DARS, pp. 285–298. Springer (2018)
15. Peshkin, L., Kim, K.E., Meuleau, N., Kaelbling, L.P.: Learning to cooperate via policy search. UAI, pp. 489–496. Morgan Kaufmann Publishers Inc. (2000)
16. Rasmussen, C.E.: Gaussian processes in machine learning. In: Summer School on Machine Learning, pp. 63–71. Springer (2003)
17. Ruiz, A.V., Xu, Z., Merino, L.: Distributed multi-robot cooperation for information gathering under communication constraints. In: ICRA, pp. 1267–1272. IEEE (2018)
18. Shieh, E.A., Jiang, A.X., Yadav, A., Varakantham, P., Tambe, M.: Unleashing decmdps in security games: Enabling effective defender teamwork. In: ECAI, vol. 263, pp. 819–824 (2014)
19. Singh, A., Krause, A., Guestrin, C., Kaiser, W.J., Batalin, M.A.: Efficient planning of informative paths for multiple robots. In: IJCAI, vol. 7, pp. 2204–2211 (2007)
20. Viseras, A., Wiedemann, T., Manss, C., Magel, L., Mueller, J., Shutin, D., Merino, L.: Decentralized multi-agent exploration with online-learning of gaussian processes. In: ICRA (2016)