

Efficient Online Multi-robot Exploration via Distributed Sequential Greedy Assignment

Micah Corah and Nathan Michael
The Robotics Institute, Carnegie Mellon University
Email: {micahcorah, nmichael}@cmu.edu

Abstract—This work addresses the problem of efficient online exploration and mapping using multi-robot teams via a distributed algorithm for planning for multi-robot exploration—distributed sequential greedy assignment (DSGA)—based on the sequential greedy assignment (SGA) algorithm. SGA permits bounds on suboptimality but requires all robots to plan in series. Rather than plan for robots sequentially as in SGA, DSGA assigns plans to subsets of robots during a fixed number of rounds. DSGA retains the same suboptimality bounds as SGA with the addition of a term describing suboptimality introduced due to redundant sensor information. We use this result to extend a single-robot planner based on Monte-Carlo tree search to the multi-robot domain and evaluate the resulting planner in simulated exploration of a confined and cluttered environment. The experimental results show that suboptimality due to redundant sensor information introduced by the distributed planning rounds remains near zero in practice when using as few as two or three distributed planning rounds and that DSGA achieves similar or better objective values and entropy reduction as SGA while providing a 2–6 times computational speedup for multi-robot teams ranging from 4 to 32 robots.

I. INTRODUCTION

We consider multi-robot exploration as the problem of actively mapping environments by planning actions for multiple agents in order to produce informative sensor measurements. In this work, we address the problem of planning for exploration with large teams of robots using distributed computation and emphasize online planning and operation in confined and cluttered environments.

Informative planning problems of this form are NP-hard [18]. Rather than attempt to find an optimal solution in possibly exponential time, we seek approximate solutions with bounded suboptimality that can be found efficiently in practice. A commonly used suboptimal planning algorithm is sequential greedy assignment (SGA) [1, 7, 27]. SGA assigns plans to robots in sequence using a single-robot planner to maximize mutual information between a robot’s future observations and the explored map given knowledge of plans already assigned to other robots. SGA-based approaches that leverage mutual information objectives for multi-robot exploration achieve a two-times suboptimality bound when comparing the approximate solution found by the greedy algorithm to the true optimum [24], and arbitrary planners with known suboptimality achieve a similar bound [28].

The sequential nature of SGA results in rapidly increasing computation time which precludes online planning for large numbers of robots. This increased computation time is

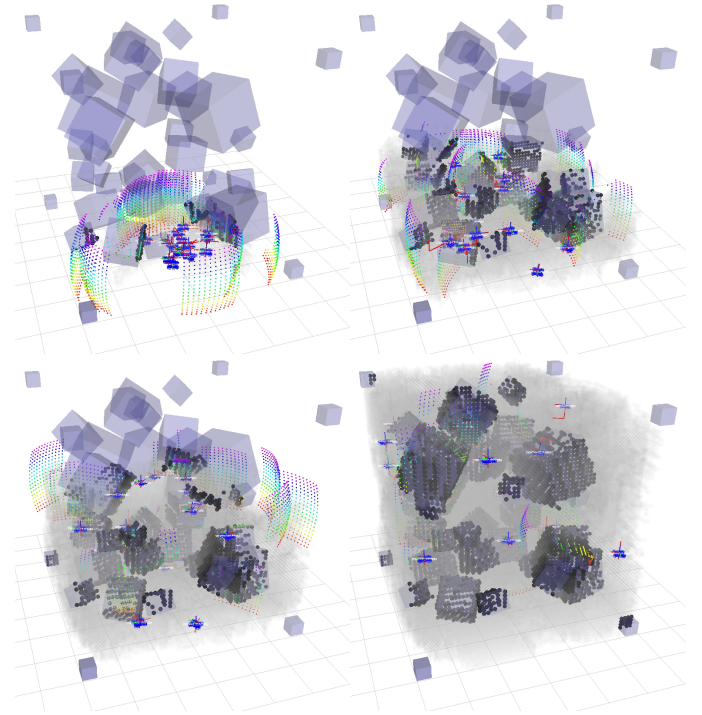


Fig. 1: An example exploration experiment. A multi-robot team explores a three-dimensional environment cluttered with numerous obstacles (cubes) while using an online distributed planner. Known empty space is gray, and occupied space is black. Robots are shown in blue with red trajectories and obtain rainbow-colored point-cloud observations from their depth-cameras. (top-left) The robots begin with randomized initial positions near a lower edge of the exploration environment which is bounded by a cube. (top-right) After entering the environment robots spread out to cover the bottom of the cubic environment (bottom-left) and then proceed upward to cover more of the volume. (bottom-right) Given enough time the robots explore the entire environment.

especially relevant to exploration problems as new information about occupancy significantly affects both feasible and optimal plans and necessitates reactive planning to achieve high rates of exploration. We propose a modified version of SGA, distributed sequential greedy assignment (DSGA), which consists of a fixed number of sequential planning rounds. At the beginning of each round, robots plan in parallel using a single-robot planner. A subset of those plans is chosen to minimize the difference between the information gain for

the entire subset and the sum of the information gains for each robot individually which does not consider redundancy between robots. We obtain a performance bound in terms of the result found by Singh et al. [28] that explicitly describes the additional suboptimality as exactly the reduction in the objective values accrued during the subset selection process. In doing so, we reduce the planning problem to analyzing and taking advantage of the coupling between robots' observations during the subset selection, to enable faster online computation in a distributed manner without compromising exploration performance.

II. RELATED WORK

Early works in robotic exploration often approach the problem through geometric methods, such as frontier-based approaches [31]. Recent approximations of mutual information for ranging sensors [9, 16] has led to the development of exploration approaches that seek to directly maximize mutual information [3, 8, 15, 19, 21, 29]. We similarly formulate exploration as finite-horizon maximization of mutual information, and the contribution of this work is to propose and analyze a new distributed algorithm for multi-robot exploration problems. The goals of this methodology are similar to those of Best et al. [3] who use probability collectives in an anytime planner.

More formally, the mutual information is submodular and nondecreasing, and the joint-space of multi-robot trajectories forms a matroid. Recent results on matroid constrained submodular maximization [5, 13] provide randomized polynomial time algorithms with $1 - 1/e$ suboptimality guarantees, improving on the previous best known guarantee of $1/2$ for SGA. This bound is tight for polynomial-time algorithms for subset selection [22, 17] which is a special case of the matroid-constraint. These approaches are computationally expensive and do not necessarily generalize well to multi-robot distributed planning. We therefore focus on SGA for the use case of online planning given its combination of reasonable bounds and runtime.

Several recent works consider the problem of developing parallel algorithms for unconstrained submodular maximization [20, 32] in addition to matroid constrained submodular maximization [2, 32]. However, achieving reasonable bounds can require additional assumptions on functional and algorithmic properties that may prove inappropriate in a distributed multi-robot context [20]. Mirzasoleiman et al. [20] propose a parallel algorithm for cardinality constrained submodular maximization, and similarly, Barbosa et al. [2] extend results by Calinescu et al. [5] to obtain a parallel algorithm applicable to matroid-constrained problems. However, these algorithms are based on data-parallel approaches that distribute planning for each individual robot across all processors. Instead, we prefer a robot-centric approach such that individual robots are ultimately responsible for planning and selecting their own actions.

III. PRELIMINARIES

Before presenting the details of the formulation and algorithm, we present some brief background details on information theory and submodular functions.

A. Information theory

Entropy quantifies the uncertainty in a random variable in terms of the average number of bits necessary to disambiguate a random variable X and denoted as

$$H(X) = \sum_i -\mathbb{P}(X = i) \log_2 \mathbb{P}(X = i), \quad (1)$$

with the entropy conditional on Y as

$$H(X|Y) = \sum_i \sum_j -\mathbb{P}(X = i, Y = j) \log_2 \mathbb{P}(X = i|Y = j). \quad (2)$$

The goal of the exploration problem is then to reduce the entropy of the map, $H(M)$. Mutual information quantifies the expected reduction of the entropy given an observation Y

$$\begin{aligned} I(X; Y) &= \sum_i \sum_j -\mathbb{P}(X = i, Y = j) \log_2 \frac{\mathbb{P}(X = i)\mathbb{P}(Y = j)}{\mathbb{P}(X = i, Y = j)} \\ &= H(X) - H(X|Y). \end{aligned} \quad (3)$$

Cover and Thomas [11] provide more detailed coverage of information theory and the properties of entropy and mutual information.

B. Submodularity for sequential greedy assignment

For conditionally independent observations, the mutual information is a submodular, nondecreasing set-function [17]. As shown by Nemhauser et al. [23, 24], these properties permit useful suboptimality bounds for greedy algorithms that we will leverage to develop an efficient algorithm for multi-robot active perception.

Define the set function, $g : 2^\Omega \rightarrow \mathbb{R}$ where 2^Ω is the power-set of the ground-set, Ω . Then g is submodular if, for any $A \subseteq B \subseteq \Omega$ and $C \subseteq \Omega \setminus B$, the following inequality holds

$$g(A \cup C) - g(A) \geq g(B \cup C) - g(B) \quad (4)$$

This function is also monotonic if for any $A \subset \Omega$ and $x \in \Omega \setminus A$

$$g(A \cup \{x\}) \geq g(A) \quad (5)$$

with $g(\emptyset) = 0$.

IV. MULTI-ROBOT EXPLORATION FORMULATION

This section describes the problem of distributed multi-robot exploration. We begin by describing the system and environment models and then introduce the planning problem as a finite-horizon optimization.

A. System model

Consider a team of robots, $R = \{r_1, \dots, r_{n_r}\}$, engaged in exploration of some environment m . The dynamics and sensing are described by

$$x_t = f(x_{t-1}, u), \quad (6)$$

$$y_t = h(x_t, m) + \nu \quad (7)$$

where x_t represents a robot's state at time, t , and $u \in \mathcal{U}$ is the control input. The observation, y_t , is a function of both the state and the environment and is corrupted by noise, ν . We use capital letters to refer to random variables and lower-case for realizations so M and Y_t represent random variables associated with the environment and an observation, respectively.

B. Occupancy grids

The environment, M , is represented as an occupancy grid Elfes [12] with an associated mutual information approximation for ranging sensors [9, 16]. The environment is discretized into cells, $M = \{C_1, \dots, C_{n_m}\}$, that are either occupied or free with some probability. Cells are independent such that the probability of a realization m is $\mathbb{P}(M = m) = \prod_{i=1}^{n_m} \mathbb{P}(C_i = c_i)$. The conditional probability of M given previous states and observations is then written

$$\begin{aligned} & \mathbb{P}(M = m | x_{1:T-1}, y_{1:T-1}) \\ &= \frac{\prod_{t=0}^T \mathbb{P}(y_{t-1} | M = m, x_{t-1}) \mathbb{P}(M = m)}{\sum_{m' \in \mathcal{M}} \prod_{t=0}^T \mathbb{P}(y_{t-1} | M = m', x_{t-1}) \mathbb{P}(M = m')} \end{aligned} \quad (8)$$

As representing an unconstrained joint distribution between cells is infeasible, the conditional probabilities of the cells given previous measurements is also treated as being independent with probability $p_{i,t}$ such that the conditional probability is

$$\mathbb{P}(M = m | x_{1:T-1}, y_{1:T-1}) = \prod_{i=1}^{n_m} p_{i,t} \quad (9)$$

We denote the collection of probabilities as the belief, $b_t = \bigcup_{i=1}^{n_m} p_{i,t}$.

C. Problem description and objective

For one robot and one time-step the optimal control action in terms of entropy reduction is

$$u_1^* = \arg \max_{u \in \mathcal{U}} I(Y_{t+1}; M | b_t). \quad (10)$$

Consider an l -step lookahead. The problem becomes a belief-dependent partially-observable Markov decision process (POMDP) as is discussed in more detail by Lauri and Ritala [19]. In the general case, this is an optimization over policies

$$Q_l(b_t, x_t, u) = I(Y_{t+1}; M | b_t) + \mathbb{E}_{Y_{t+1}} [\max_{u' \in \mathcal{U}} Q_{l-1}(b_{t+1}, x_{t+1}, u')], \quad (11)$$

$$u_l^* = \arg \max_{u \in \mathcal{U}} Q_l(b_t, x_t, u). \quad (12)$$

We instead optimize over a fixed series of actions rather than over policies which results in a simpler problem. To simplify notation and possible confusion of the relationship between observations and controls, let \mathcal{Y}_i indicate the space of possible observations available to robot i over the finite horizon induced by the control inputs and dynamics. The optimal multi-robot, finite-horizon informative plan is then

$$Y_{t+1:t+l, 1:n_r}^* = \arg \max_{Y_{1:t, 1:n_r} \in \mathcal{Y}_{1:n_r}} I(Y_{1:t, 1:n_r}; M | b_t) \quad (13)$$

where the indexing $x_{1:t, 1:n_r}$ represents values at times 1 through t and for robots 1 through n_r . In the following sections, we will drop the time and robot index when appropriate.

D. Assumptions

We make the following assumptions regarding the exploration scenario: 1) all agents have the same belief state, operate synchronously, and communicate via a fully connected network; 2) the transition function, f is bounded; and 3) the sensor range is bounded. The first assumption simplifies analysis in the context of this work. Here we emphasize scenarios where large numbers of robots operate in close proximity leading to redundant observations. Extending the proposed algorithm to incorporate additional considerations such as communication constraints is left to future work. The second and third assumptions ensure that the mutual information between observations made by distant robots is zero. These assumptions simplify the problem structure and are the key reason that the proposed efficient algorithm comes with little to no reduction in solution quality.

V. SINGLE-ROBOT PLANNING

We employ Monte-Carlo tree search [10, 4] for the single robot planner as previously proposed for active perception and exploration [19, 3, 25] and in multi-robot active perception [3].

In order to ensure bounded and similarly scaled rewards, constant terms from (13) are dropped when planning for the i^{th} robot to obtain

$$I(Y_{t+1:t+l}; M | Y_{t+1:t+l, A}, b_t) \quad (14)$$

and maximize the mutual information between $Y_{t+1:t+l}$ conditional on observations, $Y_{t+1:t+l, A}$ for the set of robots, A

Denote solutions obtained from the Monte-Carlo tree search single-robot planner maximizing (14) as

$$\hat{Y}_i = \text{SingleRobot}(i, Y_A) \quad (15)$$

and assume this planner has suboptimality $\eta \geq 1$ such that

$$\eta I(M; \hat{Y}_i | Y_A) \geq \max_{Y \in \mathcal{Y}_i} I(M; Y | Y_A) \quad (16)$$

as by Singh et al. [28].

VI. MULTI-ROBOT PLANNING

The main contribution of this work is in the design and analysis of a new distributed multi-robot planner that extends the single-robot planner discussed in Sect. V or any planner satisfying (16) to multi-robot exploration. In development of a

Algorithm 1 Distributed sequential greedy assignment (DSGA) from the perspective of robot i

```

1:  $n_d \leftarrow$  number of planning rounds
2:  $n_r \leftarrow$  number of robots
3:  $Y_F \leftarrow \emptyset$  ▷ set of fixed trajectories
4: for  $1, \dots, n_d$  do
5:    $Y_{i|Y_F} \leftarrow$  SingleRobot( $i, Y_F$ )
6:    $I_{i,0} \leftarrow I(M; Y_{i|Y_F} | Y_F)$  ▷ planner reward
7:    $I_{i,F} \leftarrow I_{i,0}$  ▷ updated reward
8:   for  $k = 1, \dots, \lceil \frac{n_r}{n_d} \rceil$  do
9:      $j \leftarrow \arg \min_{j \in \{1, \dots, n_r\}} I_{j,F} - I_{j,0}$  ▷ reduction across robots
10:    if  $i = j$  then
11:      Transmit:  $Y_{i|Y_F}$ 
12:      return  $Y_{i|Y_F}$ 
13:    else
14:      Receive:  $Y_{j|Y_F}$ 
15:       $Y_F \leftarrow Y_F \cup Y_{j|Y_F}$ 
16:       $I_{i,F} \leftarrow I(M; Y_{i|Y_F} | Y_F)$  ▷ update reward

```

distributed algorithm, we first present a commonly used algorithm, SGA, which provides suboptimality guarantees [28] but requires robots to plan sequentially. We then propose a similar distributed algorithm, DSGA, and analyze its performance in terms of time and suboptimality.

A. Sequential greedy assignment

Consider an algorithm that plans for each robot in the team maximizing (14) given all previously assigned plans and continues in this manner to sequentially assign plans to each robot. We will refer to this as sequential greedy assignment (SGA). Singh et al. [28] use the properties of mutual information discussed in Sect. III-B to establish that SGA obtains an objective value within $1 + \eta$ of the optimal solution. The greedy solution using an optimal single-robot planner can be defined inductively as $Y^g = Y_{0:n_r}^g$ using a suboptimal planner as in (15) to obtain the solution $\hat{Y}^g = \hat{Y}_{0:n_r}^g$ such that

$$\begin{aligned} \hat{Y}_0^g &= \emptyset \\ \hat{Y}_i^g &= \text{SingleRobot}(i, Y_{1:i-1}^g) \end{aligned} \quad (17)$$

This algorithm satisfies the following suboptimality bound.

Theorem 1 (Suboptimality bound of sequential assignment [28]). *SGA obtains a suboptimality bound of*

$$I(M; Y^*) \leq (1 + \eta)I(M; \hat{Y}^g) \quad (18)$$

This multi-robot planner is formulated as an extension of a generic single-robot planner and depends only on the suboptimality of the single-robot planner. As robots plan sequentially, this leads to large computation times as the number of robots grows.

B. Distributed sequential greedy assignment

Consider a scenario with spatially distributed robots such that the mutual information between any pair of robots is zero. The

union of solutions obtained for individuals independently is then equivalent to a solution to the combinatorial problem over all robots, Y^* . A weaker version of this idea applies such that if the plans returned for a subset of robots are conditionally independent, those plans are optimal over that subset of robots regardless of the inter-robot distances. The distributed planner, DSGA, is designed according to this principle and allows all robots to plan at once and then selects a subset of those plans while minimizing suboptimality.

DSGA is defined in Alg. 1 from the perspective of robot i . Planning proceeds in a fixed number of rounds, n_d (line 4). Each round begins with a planning phase where each robot plans for itself given the set of plans that are assigned in previous rounds (line 5), stores the initial objective value, $I_{i,0}$ (line 6), and copies this to a variable, $I_{i,F}$ (line 7) that represents the updated value as more plans are assigned. The round ends with a selection phase (line 8) during which a subset of $\lceil \frac{n_r}{n_d} \rceil$ plans are assigned to robots. The plans are assigned greedily to minimize the decrease in the objective values, $I_{j,F} - I_{j,0}$, and the plan to be assigned is computed using a reduction across the multi-robot team (line 9). The chosen robot sends its plan to the other robots (line 11), and these robots store this plan (line 15) and update their objective values (line 16).

Denote a planner with n_d planning rounds as DSGA_{n_d} . Let D_i be the set of robots whose trajectories are assigned during the i^{th} distributed planning round and $F_i = \bigcup_{j=1}^i D_j$ as the set of all robots with trajectories assigned by that round. Denote incremental solutions to this new distributed algorithm, similarly to the previously discussed algorithms, as $\hat{Y}_{F_i}^d$. Then, let $\hat{Y}_{r|Y_{F_i}^d}$ represent the approximate solution returned by the single-robot planner given previously assigned trajectories. The result of DSGA can then be written as $Y_{D_{i,j}}^d = \hat{Y}_{D_{i,j}|Y_{F_{i-1}}^d}$ where $D_{i,j}$ is the j^{th} robot assigned during round i . DSGA achieves a bound related to Theorem 1 with an additive term based on the decrease in objective values from initial planning to assignment that DSGA seeks to minimize (Alg. 1, line 9).

Theorem 2. *The excess suboptimality of the distributed algorithm compared to greedy sequential assignment is given by the sum of mutual information between each selection and all prior selections during that round¹*

$$I(M; Y^*) \leq (1 + \eta)I(M; Y^d) + \psi \quad (19)$$

where $\psi = \eta \sum_{i=1}^{n_d} \sum_{j=1}^{|D_i|} I(Y_{D_{i,j}}^d; Y_{D_{i,1:j-1}}^d | Y_{F_{i-1}}^d)$ is this excess suboptimality. The proof is provided in the appendix.

This is an online bound in the sense that it is parametrized by the planner solution. However, as will be shown in the results, ψ tends to be small in practice indicating that DSGA produces results comparable to SGA. In this sense, small values of ψ serve to certify the greedy bound of $1 + \eta$

¹Although this paper addresses multi-robot exploration, this result applies generally to informative planning problems and general matroid-constrained monotone submodular maximization (aside from notation and problem specialization).

empirically without needing to obtain the objective value returned by SGA explicitly. This bound can be extended to provide additional insights into and to produce algorithms that better take account for problem structure.

Corollary 2.1. *Using submodularity, excess suboptimality may be bounded by a sum over pairwise conditional mutual information rather than the updated mutual information describing the performance of subsets at the beginning of the subset selection phase*

$$I(M; Y^*) \leq (1 + \eta)I(M; Y^d) + \eta \sum_{i=1}^{n_d} \sum_{j=1}^{|D_i|} \sum_{k=1}^{j-1} I(Y_{D_{i,j}}^d; Y_{D_{i,k}}^d | Y_{F_{i-1}}^d). \quad (20)$$

By submodularity, we may also drop all conditioning to obtain a bound on any given partitioning based on the problem structure

$$I(M; Y^*) \leq (1 + \eta)I(M; Y^d) + \eta \sum_{i=1}^{n_d} \sum_{j=1}^{|D_i|} \sum_{k=1}^{j-1} I(Y_{D_{i,j}}^d; Y_{D_{i,k}}^d). \quad (21)$$

Equation (21) extends to pairwise bounds over the space of all plans by upper-bounding the terms of the summation such as with a bound based on inter-robot distances. This bound may be used to obtain algorithms with bounded performance based on the density of the deployment of the multi-robot team or to inform the subset selection process.

In DSGA, subsets are chosen using a greedy strategy. This performs well when the problem is balanced among robots and sufficiently decoupled. For unbalanced problems, a large number of robots with low objective values can cause relevant robots to be selected during later rounds, eliminating benefits of the sequential rounds. The negation of the contribution of a single round is

$$I(M; Y_{D_i}^d | Y_{F_{i-1}}^d) - \sum_{j=1}^{|D_i|} I(M; Y_{D_{i,j}}^d | Y_{F_{i-1}}^d) \quad (22)$$

found by application of the chain-rule of mutual information to (32). Equation (22) is submodular and non-increasing unlike the nondecreasing objectives considered previously. Existing bounds for this problem class provide relatively poor [14] bounds and so we continue with the greedy heuristic.

C. Algorithm runtime analysis

We compare the runtime of DSGA to SGA for variable numbers of robots. With runtime defined as the time elapsed from when the first robot begins computation until the last robot is finished. We assume point-to-point communication over a fully-connected network requiring a fixed amount of time per message. Messages have fixed sizes, corresponding to either a finite-horizon plan or a difference in mutual information. Given these assumptions, broadcast and reduction steps each require $O(\log n_r)$ time. The Monte-Carlo tree search planner is run for a fixed number of iterations, and the only

variability in runtime for this step enters through evaluation of mutual information. Using the approximation developed by Charrow et al. [9], evaluation of mutual information is linear in the number of cells of the map being observed. Given the assumption of bounded sensor range, evaluation of mutual information scales linearly in the number of robots.

SGA consists of n_r planning steps, each with a bounded number of mutual information evaluations and one broadcast step. The computation time of sequential greedy assignment is then

$$\text{SGA} : O(n_r^2 + n_r \log n_r). \quad (23)$$

Each round of DSGA begins with a single planning phase, and with n_d of such rounds, the time required for planning is $O(n_d n_r)$. The rest of the algorithm consists of subset selection, broadcast of the chosen plans, and computation of mutual information. These steps cumulatively run once per robot for a total cost of

$$\text{DSGA}_{n_d} : O(n_d n_r + n_r \log n_r + n_r^2). \quad (24)$$

Although the asymptotic runtime of these algorithms is quadratic, the constant factors vary significantly. For SGA, the squared term is associated with the single-robot planner which represents a large number of mutual information evaluations. In DSGA, the squared term corresponds to a single evaluation per robot which significantly reduces planning time in practice.

VII. RESULTS AND DISCUSSION

We evaluate the proposed approach using three experiments. In Sect. VII-B, we use a series of tests with sixteen-robot teams and up to three planning rounds ($n_d = 3$) and demonstrate that the excess, ψ , becomes insignificant given only a few planning rounds and in turn the performance of DSGA matches SGA. In Sect. VII-C, we test DSGA₃ and SGA with increasing numbers of robots (4, 8, 16, and 32) to show that entropy reduction performance of DSGA₃ consistently tracks SGA and that per-robot performance degrades gracefully for both algorithms as the environment becomes crowded with increasing numbers of robots. Section VII-D evaluates computation times and demonstrates significant improvements when using DSGA.

A. Implementation details

We evaluate the proposed algorithm in simulation and run experiments on a laptop equipped with an Intel i7-5600U CPU (2.6 GHz). Tests evaluating exploration performance are each run twenty times with randomized initializations. Timing tests are run separately over single runs with an identical experimental setup. Robots move through a 3D environment with planning, mapping, and mutual information computation in 3D.

The simulated robots emulate kinematic quadrotors moving in a three-dimensional environment. Robots execute discrete actions, translations of ± 0.3 m in the x - y - z directions and heading changes of ± 0.3 rad. Each robot is equipped with a simulated time-of-flight camera with a range of 2.4 m similar to a typical depth camera, 19×12 resolution, and $43.6 \times 34.6^\circ$

TABLE I: Exploration performance per robot-iteration (bits). Average results for exploration performance are shown over all trials. For the sake of presentation we assume an optimal single-robot planner so that $\eta = 1$

Alg.	n_r	Reward		Excess (ψ)		Bound		Entropy red.	
		avg.	std.	avg.	std.	avg.	std.	avg.	std.
DSGA ₁	16	26.6	6.43	6.01	3.60	59.3	15.0	372	124
DSGA ₂	16	27.6	6.48	1.27	1.06	56.4	13.4	375	144
SGA	4	30.5	12.7	-	-	61.0	25.4	359	264
DSGA ₃	4	30.9	12.5	0.285	1.03	62.1	25.1	368	255
SGA	8	30.2	9.75	-	-	60.3	19.5	374	188
DSGA ₃	8	30.2	9.84	0.211	0.614	60.6	19.7	383	197
SGA	16	28.1	6.97	-	-	56.2	13.9	383	147
DSGA ₃	16	28.1	7.31	0.323	0.421	56.5	14.7	382	148
DSGA ₃	16	28.1	7.31	0.323	0.421	56.5	14.7	382	148
SGA	32	26.2	7.11	-	-	52.3	14.2	328	124
DSGA ₃	32	27.9	6.76	0.897	0.530	56.7	13.7	330	116

field-of-view oriented with the long axis aligned vertically for use in sweeping motions. For efficient computation, rays are down-sampled by two for computation of mutual information and we substitute Shannon mutual information (3) with Cauchy-Schwarz mutual information in the implementation [9, 6], and rather than the typical uniform prior used in mapping, we introduce a prior of a 12.5% occupancy probability during evaluation of mutual information [29].

The planner and other components of the system are implemented using C++ and ROS [26]. For the distributed planner, timing results for the single-robot planning phase are computed by taking the maximum over each round and for information propagation as the maximum time over each assignment. In practice, computing the reduction to find the minimum excess term (Alg. 1, line 9) requires an insignificant amount of time. So, although we assume a logarithmic-time parallel reduction in the analysis, we compute this by iteration over all elements in the implementation.

a) Exploration scenario: We test the exploration methodology in a confined and cluttered environment with obstacles (cubes) of various sizes with robot positions initialized randomly near a lower edge as depicted in Fig. 1. The environment is bounded by a $6\text{ m} \times 6\text{ m} \times 6\text{ m}$ cube. The robots map this environment using a 3D occupancy grid with 0.1 m resolution. The confines and clutter ensure that robots remain proximal, leading to significant potential for redundant observations and suboptimal joint plans.

B. Different numbers of distributed planning rounds

Figure 2 and Table I show results for exploration experiments comparing DSGA₁ through DSGA₃ to SGA for a team of 16 robots. The excess (ψ) is largest at the beginning of each exploration run as all robots are initialized near the same position. As the robots spread out, all planners approach approximately steady-state conditions in terms of both excess suboptimality and objective values before decaying once the environment is mostly explored. The ψ terms remain relatively large for DSGA₁—which assigns all plans in a single round and does not consider conditional dependencies—and is, on average, approximately one-third of the mutual information

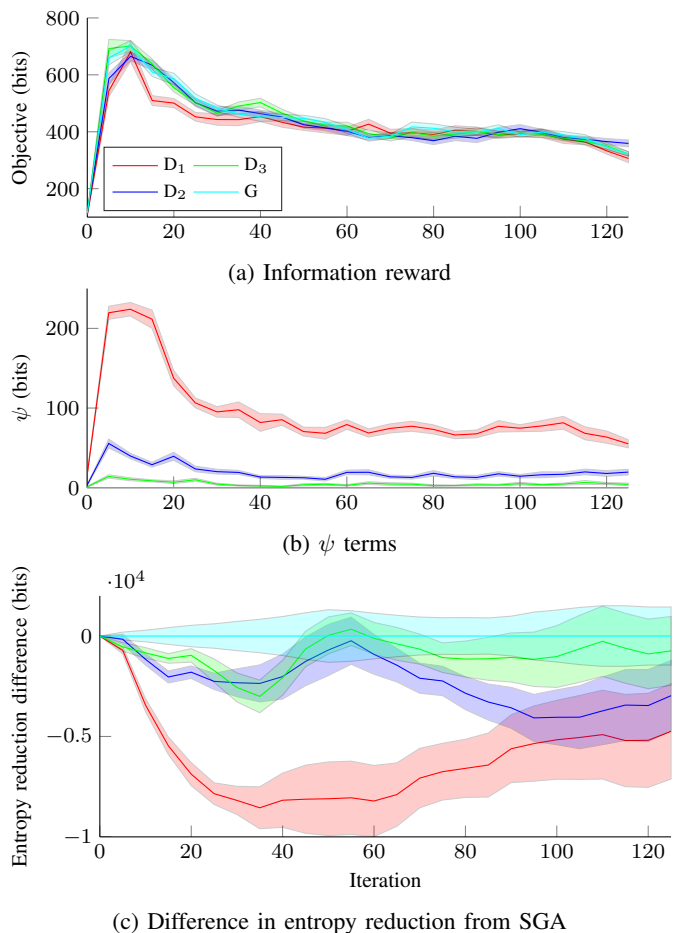


Fig. 2: Exploration results for $n_r = 16$ robots and varying the number of distributed planning rounds. (a) The mutual information objective of DSGA closely tracks SGA and improves from DSGA₁ (which ignores inter-robot interaction) to DSGA₃. (b) Similarly, the excess submodularity (ψ) terms approach zero for DSGA₃ indicating performance comparable to SGA. (c) The difference between DSGA and SGA planners in cumulative entropy reduction in the map as actual entropy reduction differs slightly. Using just $n_d = 3$ planning rounds, DSGA closely approximates and sometimes exceeds the performance of SGA which closely reflects the expected results based on the changes in objective values and excess terms. Transparent patches show standard-error.

objective. However, the ψ terms decrease monotonically with increasing numbers of planning rounds and are negligible for DSGA₃. Decreasing values of ψ are then reflected in the mutual information objective, whereas DSGA₂ and DSGA₃ closely track SGA while objective values for DSGA₁ are at times decreased (Fig. 2a). Theorem 2 states that if ψ is small DSGA obtains the same performance bounds as SGA. Although either algorithm may perform significantly better than these bounds the matching performance bounds, are accompanied by comparable objective values in the experimental results.

Further, the actual exploration performance in terms of entropy reduction also improves for DSGA₂ and DSGA₃ with similar performance to SGA (Fig. 2c) as expected according

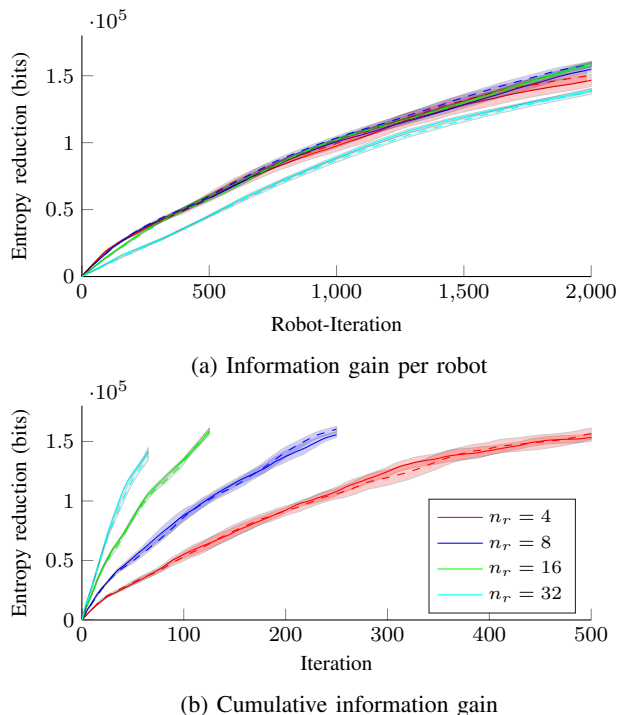


Fig. 3: Exploration performance with different numbers of robots for (solid lines) SGA and (dashed lines) DSGA₃. Note that DSGA₃ closely tracks SGA, and results sometimes appear as one line. DSGA meets or exceeds the performance of SGA for even relatively large numbers of robots despite using a constant number of planning rounds. (a) Results by robot-iteration (i.e. total robot-time). (b) Results by number of iterations (i.e. real-time). Transparent patches show standard-error.

to Fig. 2a. DSGA₁ still performs well in terms of both objective values and exploration performance despite the lack of inter-robot coordination. This motivates further study into the structure of typical informative planning problems to better explain the performance of approximate algorithms such as DSGA and the special case of DSGA₁.

C. Different numbers of robots

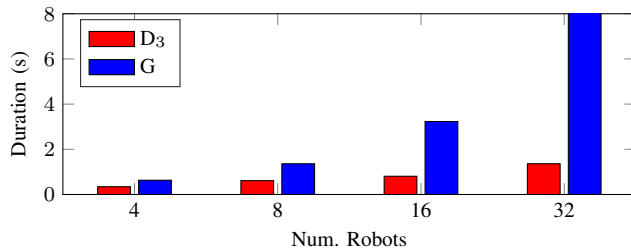
Figure 3 compares DSGA₃ to SGA for various numbers of robots first by: 1) total number of iterations to quantify the reduction in total exploration time when using large teams of robots, and 2) by number of “robot-iterations” to demonstrate that the average per-robot performance remains consistent as the number of robots increases. These results and those for varying the n_D are summarized in Table I. Robots performs consistently through $n_r = 16$ robots with a slight 10% reduction in entropy-reduction performance due to crowding with 32 robots (Fig. 3a). This consistent single-robot performance translates to a significant increase in the rate of exploration when introducing additional robots (Fig. 3b).

D. Computational performance

Figure 4 shows timing results for configurations discussed in the prior subsections. SGA scales super-linearly as expected

Alg.	n_r	S.R. Planning		Prop.		Total	
		avg.	std.	avg.	std.	avg.	std.
DSGA ₁	16	0.166	0.0214	0.176	0.0208	0.343	0.0376
DSGA ₂	16	0.407	0.0415	0.175	0.0190	0.582	0.0581
SGA	4	0.611	0.0780	0.0197	0.00307	0.631	0.0801
DSGA ₃	4	0.324	0.0311	0.0195	0.00231	0.343	0.0321
SGA	8	1.32	0.116	0.0406	0.00391	1.36	0.120
DSGA ₃	8	0.558	0.0626	0.0542	0.00625	0.612	0.0669
SGA	16	3.14	0.312	0.0904	0.00904	3.23	0.320
DSGA ₃	16	0.645	0.0737	0.162	0.0180	0.807	0.0889
SGA	32	8.00	1.18	0.220	0.0326	8.23	1.21
DSGA ₃	32	0.872	0.141	0.492	0.0723	1.36	0.202

(a) Table of timing data



(b) Timing for SGA and DSGA₃

Fig. 4: Computational performance (seconds) in terms of total computation time (time elapsed from when the first robot starts planning until the last robot stops). (a) Time per iteration spent in the single robot planner, propagation of the information reward (DSGA only), and total computation time. (b) Comparison of the timing differences between SGA and DSGA₃.

given the quadratic runtime (23). The average computation time increases nearly thirteen times from $n_r = 4$ to $n_r = 32$ robots from 0.631 s to 8.23 s. DSGA performs more reasonably both in number of planning rounds and number of robots. With sixteen robots, from DSGA₁ to DSGA₃, the computation time increases by only a factor of 2.4 despite tripling the number of planning rounds, in part due to decreased time propagating mutual information. When varying the number of robots, the average computation time varies from 0.343 s to 1.36 s leading to a 2–6 times speedup as to SGA. Time spent in the single-robot planner scales approximately linearly with the number of robots as expected and varies by relatively little from 0.324 s to 0.872 s. As expected, the information propagation scales quadratically (24) and becomes significant only with large numbers of robots. DSGA then significantly reduces the impact of the single-robot planner on computational performance as the quadratic runtime of the distributed planner is due to a single evaluation of mutual information per robot rather than many evaluations computed within the single-robot planners. This results in computation times appropriate for online planning that scale well to large numbers of robots.

VIII. CONCLUSIONS AND FUTURE WORK

The proposed distributed algorithm (DSGA) efficiently approximates sequential greedy assignment (SGA) and is appropriate for implementation on multi-robot teams using distributed computation and online planning. We apply this algorithm to the problem of multi-robot exploration, and demonstrate consistent entropy reduction performance in sim-

ulation for large numbers of robots exploring and mapping a complex three-dimensional environment. The results for DSGA demonstrate the effectiveness of this simple and efficient extension of SGA to distributed contexts by taking advantage of parallel computation. We expect that this result will be instrumental in development of physical multi-robot systems that take advantage of online distributed computation for exploration and similar finite horizon informative planning problems.

Although planning is no longer entirely sequential, the assignment in the subset selection step is still sequential resulting in the same asymptotic run time as sequential greedy assignment. Introducing bounds on the pairwise mutual information and assumptions on the multi-robot team's geometry can potentially lead to further reduction in the runtime of the proposed algorithm and extension of the proposed approach to teams of robots that have incomplete connectivity.

APPENDIX

Proof of Theorem 2: The proof of the suboptimality bound on DSGA is similar to [30] or [1] and incorporates suboptimality of the single-robot planner [28]

$$I(M; Y^*) \leq I(M; Y^*) + \sum_{i=1}^{n_r} \left(I(M; Y_{F_{n_d,1:i}}^d, Y_{F_{n_d,i+1:n_r}}^*) - I(M; Y_{F_{n_d,1:i-1}}^d, Y_{F_{n_d,i+1:n_r}}^*) \right) \quad (25)$$

$$= I(M; Y^d) + \sum_{i=1}^{n_r} \left(I(M; Y_{F_{n_d,1:i-1}}^d, Y_{F_{n_d,i:n_r}}^*) - I(M; Y_{F_{n_d,1:i-1}}^d, Y_{F_{n_d,i+1:n_r}}^*) \right) \quad (26)$$

$$\leq I(M; Y^d) + \sum_{i=1}^{n_r} \left(I(M; Y_{F_{n_d,1:i-1}}^d, Y_{F_{n_d,i}}^*) - I(M; Y_{F_{n_d,1:i-1}}^d, Y_{F_{n_d,i}}^*) \right) \quad (27)$$

$$= I(M; Y^d) + \sum_{i=1}^{n_d} \sum_{j=1}^{|D_i|} I(M; Y_{D_{i,j}}^* | Y_{D_{i,1:j-1} \cup F_{i-1}}^d) \quad (28)$$

$$\leq I(M; Y^d) + \sum_{i=1}^{n_d} \sum_{j=1}^{|D_i|} I(M; Y_{D_{i,j}}^* | Y_{F_{i-1}}^d) \quad (29)$$

$$\leq I(M; Y^d) + \eta \sum_{i=1}^{n_d} \sum_{j=1}^{|D_i|} I(M; \hat{Y}_{D_{i,j} | F_{i-1}} | Y_{F_{i-1}}^d) \quad (30)$$

$$= I(M; Y^d) + \eta \sum_{i=1}^{n_d} \sum_{j=1}^{|D_i|} \left(I(M; Y_{D_{i,j}}^d | Y_{F_{i-1}}^d) - I(M; Y_{D_{i,j}}^d | Y_{D_{i,1:j-1} \cup F_{i-1}}^d) + I(M; Y_{D_{i,j}}^d | Y_{D_{i,1:j-1} \cup F_{i-1}}^d) \right) \quad (31)$$

$$= (1 + \eta) I(M; Y^d) + \eta \sum_{i=1}^{n_d} \sum_{j=1}^{|D_i|} \left(I(M; Y_{D_{i,j}}^d | Y_{F_{i-1}}^d) - I(M; Y_{D_{i,j}}^d | Y_{D_{i,1:j-1} \cup F_{i-1}}^d) \right) \quad (32)$$

$$= (1 + \eta) I(M; Y^d) + \eta \sum_{i=1}^{n_d} \sum_{j=1}^{|D_i|} \left(I(Y_{D_{i,j}}^d; Y_{D_{i,1:j-1}}^d | Y_{F_{i-1}}^d) - I(Y_{D_{i,j}}^d; Y_{D_{i,1:j-1}}^d | M, Y_{F_{i-1}}^d) \right) \quad (33)$$

$$= (1 + \eta) I(M; Y^d) + \eta \sum_{i=1}^{n_d} \sum_{j=1}^{|D_i|} I(Y_{D_{i,j}}^d; Y_{D_{i,1:j-1}}^d | Y_{F_{i-1}}^d) \quad (34)$$

Equation (25) follows from monotonicity of mutual information and is rearranged to obtain (26). Equation (27) follows from submodularity. Equation (28) rewrites the previous expression using conditional mutual information as a sum over the planning rounds. Equation (29) again follows from submodularity. Equation (30) follows from substitution of (16). In Equation (31), we introduce terms that sum to zero, and in (32) we extract one of these terms using a telescoping sum. We now have a result that expresses the excess suboptimality in terms of the decrease in reward from when the planner is first run to when the plan is assigned. Equation (33) rearranges the mutual information terms and follows from writing the mutual informations as differences of entropies. The last mutual information term is zero due to conditional independence of observations given the environment leading to the final result in (34). ■

ACKNOWLEDGMENTS

We gratefully acknowledge support from ARL grant W911NF-08-2-0004.

REFERENCES

- [1] N. A. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas. Decentralized active information acquisition: Theory and application to multi-robot SLAM. In *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Seattle, WA, May 2015.
- [2] R. d. P. Barbosa, A. Ene, H. L. Nguyen, and J. Ward. A new framework for distributed submodular maximization. In *Proc. of the IEEE Annu. Symp. Found. Comput. Sci.*, New Brunswick, NJ, Oct. 2016.
- [3] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch. Decentralised Monte Carlo tree search for active perception. In *Algorithmic Found. Robot.*, San Francisco, CA, Dec. 2016.

- [4] C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of Monte Carlo tree search methods. *IEEE Trans. on Comput. Intell. and AI in Games*, 4(1):1–43, 2012.
- [5] G. Calinescu, C. Chekuri, M. Pal, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011.
- [6] B. Charrow. *Information-Theoretic Active Perception for Multi-Robot Teams*. PhD thesis, University of Pennsylvania, 2015.
- [7] B. Charrow, V. Kumar, and N. Michael. Approximate representations for multi-robot control policies that maximize mutual information. *Auton. Robots*, 37(4):383–400, 2014.
- [8] B. Charrow, G. Kahn, S. Patil, S. Liu, K. Goldberg, P. Abbeel, N. Michael, and V. Kumar. Information-theoretic planning with trajectory optimization for dense 3D mapping. In *Proc. of Robot.: Sci. and Syst.*, Rome, Italy, July 2015.
- [9] B. Charrow, S. Liu, V. Kumar, and N. Michael. Information-theoretic mapping using Cauchy-Schwarz quadratic mutual information. In *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Seattle, WA, May 2015.
- [10] G. Chaslot. *Monte-Carlo Tree Search*. PhD thesis, Universiteit Maastricht, 2010.
- [11] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, NY, 2012.
- [12] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer Society*, 22(6):46–57, 1989.
- [13] Y. Filmus and J. Ward. A tight combinatorial algorithm for submodular maximization subject to a matroid constraint. In *Proc. of the IEEE Annu. Symp. Found. Comput. Sci.*, New Brunswick, NJ, Oct. 2012.
- [14] S. O. Gharan and J. Vondrák. Submodular maximization by simulated annealing. Jan. 2011.
- [15] M. G. Jadidi, J. V. Miro, and G. Dissanayake. Mutual information-based exploration on continuous occupancy maps. In *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Hamburg, Germany, Sept. 2015.
- [16] B. J. Julian, S. Karaman, and D. Rus. On mutual information-based control of range sensing robots for mapping applications. *Intl. Journal of Robotics Research*, 33(10):1357–1392, 2014.
- [17] A. Krause and C. E. Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proc. of the Conf. on Uncertainty in Artif. Intell.*, Edinburgh, Scotland, 2005.
- [18] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, 9:235–284, 2008.
- [19] M. Lauri and R. Ritala. Planning for robotic exploration based on forward simulation. *Robot. Auton. Syst.*, 83, 2016.
- [20] B. Mirzasoleiman, R. Sarkar, and A. Krause. Distributed submodular maximization: Identifying representative elements in massive data. In *Adv. in Neural Inf. Process. Syst.*, Stateline, Nevada, Dec. 2013.
- [21] E. Nelson and N. Michael. Information-theoretic occupancy grid compression for high-speed information-based exploration. In *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Hamburg, Germany, Sept. 2015.
- [22] G. L. Nemhauser and L. A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of operations research*, 3(3):177–188, 1978.
- [23] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions-I. *Math. Program.*, 14(1):265–294, 1978.
- [24] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions-II. *Polyhedral Combinatorics*, 8:73–87, 1978.
- [25] T. Patten. *Active Object Classification from 3D Range Data with Mobile Robots*. PhD thesis, The University of Sydney, 2017.
- [26] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, Kobe, Japan, May 2009.
- [27] T. Regev and V. Indelman. Multi-robot decentralized belief space planning in unknown environments via efficient re-evaluation of impacted paths. In *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Daejeon, Korea, Oct. 2016.
- [28] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser. Efficient informative sensing using multiple robots. *J. Artif. Intell. Res.*, 34:707–755, 2009.
- [29] W. Tabib, M. Corah, N. Michael, and R. Whittaker. Computationally efficient information-theoretic exploration of pits and caves. In *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Daejeon, Korea, Oct. 2016.
- [30] J. L. Williams. *Information Theoretic Sensor Management*. PhD thesis, Massachusetts Institute of Technology, 2007.
- [31] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proc. of the Intl. Sym. on Comput. Intell. in Robot. and Autom.*, Monterey, CA, July 1997.
- [32] T. Zhou, H. Ouyang, Y. Chang, J. Bilmes, and C. Guestrin. Scaling submodular maximization via pruned submodularity graphs. *Proc. Mach. Learn. Res.*, 54:316–324, 2017.