Autonomous Robots
https://doi.org/10.1007/s10514-018-9778-6

Distributed matroid-constrained submodular maximization for multi-robot exploration: theory and practice

Micah Corah[1] · **Nathan Michael**[1]

Received: 12 December 2017 / Accepted: 21 June 2018

Abstract
This work addresses the problem of efficient online exploration and mapping using multi-robot teams via a new distributed algorithm for multi-robot exploration, distributed sequential greedy assignment (DSGA), which is based on sequential greedy assignment (SGA). While SGA permits bounds on suboptimality, robots must execute planning steps sequentially. Rather than plan for each robot sequentially as in SGA, DSGA assigns plans to subsets of robots using a fixed number of sequential planning rounds. DSGA retains the same suboptimality bounds as SGA with the addition of a term that describes the additional suboptimality incurred when assigning multiple plans at once. 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 for teams of 4–32 robots suboptimality due to redundant sensor information introduced in the distributed planning rounds remains small in practice given only two or three distributed planning rounds while providing a 2–8 times speedup over SGA. We also incorporate aerial robots with inter-robot collision constraints and non-trivial dynamics and address subsequent impacts on safety and optimality. Real-time simulation and experimental results for teams of quadrotors demonstrate online planning for multi-robot exploration and indicate that collision constraints have limited impacts on exploration performance.

Keywords Multi-robot · Exploration · Informative planning · Submodular · Matroid

1 Introduction

We pose multi-robot exploration as the problem of actively mapping environments by planning actions for a team of sensor-equipped robots to maximize 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.

The authors gratefully acknowledge the support of ARL Grant W911NF-08-2-0004.

This is one of several papers published in *Autonomous Robots* comprising the "Special Issue on Robotics Science and Systems".

✉ Micah Corah
micahcorah@cmu.edu

Nathan Michael
nmichael@cmu.edu

[1] The Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 12513, USA

Informative planning problems of this form are known to be NP-Hard (Krause et al. 2008). 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) (Atanasov et al. 2015; Charrow et al. 2014; Regev and Indelman 2016). 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 (Nemhauser et al. 1978b), and arbitrary planners with known suboptimality achieve a similar bound (Singh et al. 2009).

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 espe-

Published online: 19 July 2018

cially 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. (2009) that explicitly describes the additional suboptimality as 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.

DSGA, suboptimality analysis, and initial experiments appeared in an earlier version of this paper (Corah and Michael 2017). This version introduces more direct treatment of the problem structure and combinatorial constraints, addresses how this structure varies under different assumptions, and provides expanded results. Worst-case analysis of DSGA is now included in addition to earlier analysis relating the performance of DSGA to SGA. Inter-robot collision constraints and related algorithmic modifications have been introduced in order to guarantee that the planner produces a collision-free joint plan after each planning step. This result enables application to physical systems and so additional results are included that demonstrate collision-free exploration for teams of quadrotors both in simulation and experimentally in a motion capture arena.

2 Related work

Early works in robotic exploration often approach the problem through geometric methods, such as frontier-based approaches (Yamauchi 1997). Recent approximations of mutual information for ranging sensors (Charrow et al. 2015b; Julian et al. 2014) have led to the development of exploration approaches that seek to directly maximize mutual information (Best et al. 2016; Charrow et al. 2015a; Jadidi et al. 2015; Lauri and Ritala 2016; Nelson and Michael 2015; Tabib et al. 2016). 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. (2016) who use probability collectives in an any-time planner.

More formally, mutual information is submodular and non-decreasing, and the joint space of multi-robot trajectories forms a matroid which can be though of as a generalization of vector spaces to sets. Recent results on matroid-constrained submodular maximization (Calinescu et al. 2011; Filmus and Ward 2012) provide randomized polynomial-time algorithms with $1 - 1/e$ suboptimality guarantees, improving on the previous best known guarantee of $1/2$ for SGA. This $1 - 1/e$ suboptimality bound is tight for polynomial-time algorithms for cardinality-constrained problems (Nemhauser and Wolsey 1978). Because a cardinality constraint is a special case of a matroid constraint (a uniform matroid) this bound is also tight for general matroids. A similar result holds for mutual information with conditionally independent observations (Krause and Guestrin 2005) for general graphical models unless $P = NP$. However, the approaches that achieve this tighter bound 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 run time.

Several recent works consider the problem of developing parallel algorithms for cardinality-constrained submodular maximization (Mirzasoleiman et al. 2013; Zhou et al. 2017) in addition to matroid-constrained submodular maximization (Barbosa et al. 2016; Zhou et al. 2017). However, achieving reasonable bounds can require additional assumptions on functional and algorithmic properties that may prove inappropriate in a distributed multi-robot context (Mirzasoleiman et al. 2013). Mirzasoleiman et al. (2013) propose a parallel algorithm for cardinality-constrained submodular maximization, and similarly, Barbosa et al. (2016) extend results by Calinescu et al. (2011) to obtain a parallel algorithm applicable to matroid-constrained problems. However, these algorithms are based on data-parallel approaches that distribute the ground set across processors. In the context of this work, distribution across processors corresponds to distribution across robots which quickly becomes intractable for large numbers of robots and when the ground set changes across problem instances. Instead, we prefer a robot-centric approach in which individual robots are responsible for planning and selecting their own actions.

Prior works in robotics that apply matroid-constrained submodular maximization (Atanasov et al. 2015; Singh et al. 2009) frequently apply suboptimality results (Goundan and Schulz 2007) for suboptimal solutions to path planning subproblems in conjunction with the locally greedy heuristic for partition matroids (SGA) (Nemhauser et al. 1978b). Recent works in robotics and control address a more broad variety of objectives and combinatorial constraints (Jorgensen et al. 2017; Williams et al. 2017). Jorgensen et al. (2017) consider

a variant of multi-agent orienteering and combine sequential assignment and suboptimal planning problems with the addition of a variety of matroid constraints such as to represent constraints on the spatial distribution of robots and resource constraints as in usage of runways. In addition, they provide explicit suboptimality bounds via problem-specific application of a suboptimal orienteering solver. Williams et al. (2017) propose a task-assignment formulation based on intersections of matroids and propose a distributed algorithm that generalizes the approach proposed by Choi et al. (2009). Although, the distributed auctions used by Williams (2007), and Choi et al. (2009) can be applied to the problems discussed in this work, they address a more general class of submodular maximization problems and incur increased asymptotic computation times.

A few works have begun to entertain distributed computation models where blocks of a partition matroid are associated with computational units and apply the locally greedy heuristic along with constraints on communication. Specifically, Gharesifard and Smith (2017) model distributed variants of SGA where agents have limited information about other agents' decisions according to a directed acyclic graph. The worst-case analysis by Gharesifard and Smith (2017) parallels results by Mirzasoleiman et al. (2013) and has recently been extended by Grimsman et al. (2017) who provide much tighter results in the same setting. While the worst case bounds in these works are directly relevant to our proposed approach, these bounds also scale with the reciprocal of the number of robots. This indicates that strong performance bounds for submodular functions and scalability to large numbers of agents can be conflicting goals. Instead, we focus on development of an approach that can take advantage of aspects of problem structure beyond submodularity, such as spatial locality, to efficiently provide distributed solutions with minimal impact on solution quality.

3 Preliminaries

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

3.1 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 is denoted as

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

with the entropy conditional on Y as

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

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

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

Cover and Thomas (2012) provide more detailed coverage of information theory and the properties of entropy and mutual information.

3.2 Submodularity for sequential greedy assignment

For conditionally independent observations, the mutual information is a submodular, non-decreasing set function (Krause and Guestrin 2005). As shown by Nemhauser et al. (1978a, b), 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, $g : 2^\Omega \to \mathbb{R}$ 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

$$g(A \cup C) - g(A) \geq g(B \cup C) - g(B). \tag{4}$$

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

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

and normalized if $g(\emptyset) = 0$.

3.3 Matroids and independence systems

The multi-robot planning problem that is the focus of this paper can be formulated as submodular maximization with a partition matroid constraint. This special case of a matroid can be optimized with constant suboptimality by locally maximizing the objective sequentially over each partition (Nemhauser et al. 1978b).

Toward this end, the tuple $(\Omega, \mathcal{I})$ where $\mathcal{I}$ is composed of subsets of the ground set such that $I \in \mathcal{I} \to I \subseteq \Omega$ is an *independence system* if $\mathcal{I}$ satisfies a heredity property such that $\emptyset \in \mathcal{I}$ and for all $I_1 \in \mathcal{I}$ then

$$I_2 \subseteq I_1 \rightarrow I_2 \in \mathcal{I}.$$

An independence system is a *matroid* if for all $I, I' \in \mathcal{I}$ such that $|I| > |I'|$ there exists some $x \in I \setminus I'$ such that

$$I' \cup \{x\} \in \mathcal{I} \tag{6}$$

which is referred to as the exchange property. A consequence of the exchange property is that all maximal independent sets have the same cardinality which is referred to as the rank of the matroid and in this work also corresponds to the number of robots. A *partition matroid* is a special case of a matroid. Consider a partitioning of the ground set into blocks $A_i \subseteq \mathcal{I}$ for all $i \in \{1 \ldots n\}$ such that $A_i \cap A_j = \emptyset$ for all $i \neq j$. Then $(\Omega, \mathcal{I})$ is a partition matroid if

$$\mathcal{I} = \{I \subseteq \Omega \mid |I \cap A_i| \leq a_i, \quad \forall i \in \{1 \ldots n\}\}$$

where $a_i \in \mathbb{Z}_+$ is the maximum number of elements that can be selected from a given block.

When $a_i = 1$ for all i, the partition matroid will be referred to as a *simple partition matroid*. These matroids will be used to model joint action spaces of multi-robot teams whereas A_i is the space of trajectories or actions associated with robot i and there are no inter-robot constraints on action selection. Later, independence systems will be recalled in discussion of inter-robot collision constraints as the partition matroid model no longer applies.

4 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.

4.1 System model

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

$$x_t = f(x_{t-1}, u), \tag{7}$$
$$y_t = h(x_t, m) + v \tag{8}$$

where x_t represents a robot's state at time $t \in \mathbb{Z}_+$ and $u \in \mathcal{U}$ is a control input selected from the finite set $\mathcal{U}$. For aerial robots as studied in this paper $x_t \in \mathbb{SE}(3)$. The observation, y_t, is a function of both the state and the environment and is corrupted by noise, v. We use capital letters to refer to random variables and lowercase for realizations so M and Y_t represent random variables associated with the environment and an observation, respectively.

4.2 Occupancy grids and ranging measurements

The environment itself is unknown and is represented by the random variable M which is modeled as an occupancy grid (Elfes 1989) and has associated approximations of mutual information for ranging sensors (Charrow et al. 2015b; Julian et al. 2014). This occupancy grid is in turn discretized into cells, $M = \{C_1, \ldots, C_{n_m}\}$, that are either occupied or free with some probability. Cell occupancy is independent such that the probability of a realization m is $\mathbb{P}(M = m) = \prod_{i=1}^{n_m} \mathbb{P}(C_i = c_i)$ whereas $c_i \in \{0, 1\}$. The conditional probability of M given previous states and observations is then written

$$\mathbb{P}(M = m | x_{1:T}, y_{1:T}) = \frac{\prod_{t=1}^{T} \mathbb{P}(y_t | M = m, x_t) \mathbb{P}(M = m)}{\sum_{m' \in \mathcal{M}} \prod_{t=1}^{T} \mathbb{P}(y_t | M = m', x_t) \mathbb{P}(M = m')} \tag{9}$$

where $\mathcal{M} \in \{0, 1\}^{n_m}$ is the set of possible environments. As representing an unconstrained joint distribution between cells is intractable, the conditional probabilities of the cells given previous observations are also treated as being independent with probability $p_{i,t}$ such that the conditional probability is

$$\mathbb{P}(M = m | x_{1:T}, y_{1:T}) = \prod_{i=1}^{n_m} p_{i,t}. \tag{10}$$

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

The robots considered in this work are equipped with depth cameras, a form of ranging sensor. Each sensor observation consists of a collection of ranging observations which each are modeled as the distance along a ray from the sensor origin to the nearest occupied cell in the environment and corrupted by additive Gaussian noise. For more detail, we refer the reader to work by Charrow et al. (2015b).

4.3 Problem description and objective

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

$$u_1^* = \arg\max_{u \in \mathcal{U}} \; I\left(Y_{t+1}; M | b_t, x_t\right) \tag{11}$$

subject to the dynamics (7) and observation model (8). 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

(2016). In the general case, this is an optimization over policies

$$Q_l(b_t, x_t, u) = I(Y_{t+1}; M|b_t) + \mathbb{E}_{Y_{t+1}}\left[\max_{u' \in U} Q_{l-1}(b_{t+1}, x_{t+1}, u')\right], \quad (12)$$

$$u_l^* = \arg\max_{u \in U} Q_l(b_t, x_t, u) \quad (13)$$

whereas $Q_0 = 0$.

We instead optimize over a fixed series of actions rather than over policies which results in a simpler problem. To simplify notation, let $\mathcal{Y}_i$ indicate the space of possible observations available to robot i over the finite horizon induced by the finite set of control inputs, the dynamics (7), and the observation model (8). The optimal multi-robot, finite-horizon informative plan is then

$$Y^*_{t+1:t+l,1:n_r} = \arg\max_{Y_{1:l,1:n_r} \in \mathcal{Y}_{1:n_r}} I\left(Y_{1:l,1:n_r}; M|b_t, x_{t,1:n_r}\right) \quad (14)$$

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 as well as robot states and belief when appropriate.

Solutions to (14) will be constructed incrementally using greedy algorithms. Using the definitions in Sect. 3.3, the constraints in (14) can be interpreted as a simple partition matroid where $\Omega = \prod_{i=1}^{n_r} \mathcal{Y}_i$ and $\mathcal{I} = \{I \in \Omega \mid |I \cap \mathcal{Y}_i| \leq 1, \ \forall i \in \mathcal{R}\}$.

4.4 Assumptions

We make the following assumptions regarding the exploration scenario: *1) all robots 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.

5 Single-robot planning

We employ Monte-Carlo tree search (Browne et al. 2012; Chaslot 2010) for the single-robot planner as previously proposed for active perception and exploration (Best et al. 2016; Lauri and Ritala 2016; Patten 2017) and in multi-robot active perception (Best et al. 2016).

In order to ensure bounded and similarly scaled rewards, constant terms from (14) are dropped when planning for the ith robot to obtain a local objective

$$I(Y_{t+1:t+l,i}; M|Y_{t+1:t+l,A}), \quad (15)$$

the mutual information between $Y_{t+1:t+l,i}$ and the map conditional on observations $Y_{t+1:t+l,A}$ obtained by some set of robots A such that $A \cap \{i\} = \emptyset$.

Denote solutions obtained from the Monte-Carlo tree search single-robot planner maximizing (15) as

$$\hat{Y}_i = \text{SingleRobot}(i, Y_A) \quad (16)$$

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

$$\eta I(M; \hat{Y}_i|Y_A) \geq \max_{Y \in \mathcal{Y}_i} I(M; Y|Y_A) \quad (17)$$

as in to the approach of Singh et al. (2009).

Although Monte-Carlo tree search does not come with a suboptimality guarantee, some existing algorithms for informative path planning (Chekuri and Martin 2005; Singh et al. 2009) do. Monte-Carlo tree search is used instead on account of limited computation time and later for ease in incorporation of inter-robot collision constraints which are time-varying.

6 Multi-robot planning

The main contribution of this work is the design and analysis of a new distributed multi-robot planner that extends the single-robot planner discussed in Sect. 5 or any planner satisfying (17) to multi-robot exploration. In development of a distributed algorithm, we first present a commonly used algorithm, SGA, which provides suboptimality guarantees (Singh et al. 2009) but requires robots to plan sequentially. We then propose a similar distributed algorithm, DSGA, and analyze its performance in terms of computation time and suboptimality.

6.1 Sequential greedy assignment

Consider an algorithm that plans for each robot in the team by maximizing (15) 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. (2009) use the properties of mutual information discussed in Sect. 3.2 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 $Y^g = Y^g_{0:n_r}$ using a suboptimal planner as in (16) to obtain the solution $Y^g = Y^g_{0:n_r}$ such that

$$\begin{aligned} Y^g_0 &= \emptyset \\ Y^g_i &= \text{SingleRobot}(i, Y^g_{1:i-1}). \end{aligned} \tag{18}$$

This algorithm satisfies the following suboptimality bound.

Theorem 1 (Suboptimality bound of sequential assignment (Singh et al. 2009)) *SGA obtains a suboptimality bound of*

$$I(M; Y^*) \leq (1+\eta) I(M; Y^g) \tag{19}$$

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.

6.2 Distributed sequential greedy assignment

Consider a scenario with spatially distributed robots such that the mutual information between any observations obtainable within a finite horizon by 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 Algorithm 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 robot whose plan is to be assigned is selected using a distributed reduction across the multi-robot team (line 9) (Ladner and Fischer 1980), and ties are broken arbitrarily. 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 ith distributed planning round and let $F_i = \bigcup_{j=1}^{i} D_j$ be the set of all robots with trajectories assigned by that round.

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** $1, \ldots, n_d$ **do**
5: $\quad Y_{i|Y_F} \leftarrow \text{SingleRobot}(i, Y_F)$
6: $\quad I_{i,0} \leftarrow I(M; Y_{i|Y_F} | Y_F)$ ▷ planner reward
7: $\quad I_{i,F} \leftarrow I_{i,0}$ ▷ updated reward
8: $\quad$ **for** $k = 1, \ldots, \lceil \frac{n_r}{n_d} \rceil$ **do**
9: $\quad\quad j \leftarrow \arg\min_{j \in 1:n_r} I_{j,0} - I_{j,F}$ ▷ computed by distributed reduction across robots
10: $\quad\quad$ **if** $i = j$ **then**
11: $\quad\quad\quad$ Transmit: $Y_{i|Y_F}$
12: $\quad\quad\quad$ **return** $Y_{i|Y_F}$
13: $\quad\quad$ **else**
14: $\quad\quad\quad$ Receive: $Y_{j|Y_F}$
15: $\quad\quad\quad Y_F \leftarrow Y_F \cup Y_{j|Y_F}$
16: $\quad\quad\quad I_{i,F} \leftarrow I(M; Y_{i|Y_F} | Y_F)$ ▷ update reward

Denote solutions to this new distributed algorithm as Y^d. Then, let $\hat{Y}_{r|Y^d_{F_i}}$ represent the approximate solution returned by the single-robot planner given previously assigned trajectories. The result of DSGA can then be written as $Y^d_{D_{i,j}} = \hat{Y}_{D_{i,j}|Y^d_{F_{i-1}}}$ where $D_{i,j}$ is the jth 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 (Algorithm 1, line 9).

Theorem 2 *The excess suboptimality of the distributed algorithm (Algorithm 1) compared to the bound for greedy sequential assignment is given by the sum of conditional mutual informations between each selected observation and prior selections for the same planning round*[1,2]

$$I(M; Y^*) \leq (1+\eta) I(M; Y^d) + \eta\psi \tag{20}$$

where $\psi = \sum_{i=1}^{n_d} \sum_{j=1}^{|D_i|} I(Y^d_{D_{i,j}}; Y^d_{D_{i,1:j-1}} | Y^d_{F_{i-1}})$ *represents excess suboptimality. The proof is provided in Appendix A.*

This is an online bound in the sense that it is parametrized by the planner solution. However, as will be shown in the

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

[2] A previous version of this paper (Corah and Michael 2017) included a corollary to Theorem 2 that mistakenly applied submodularity to ψ in an attempt to provide bounds that could be evaluated at different stages of the planning process. Although this bound does not hold, we note that none of the rest of this work depends on this result. However, unlike as for mutual information, some common submodular functions such as set and area coverage have higher-order properties that are related to submodularity and are amenable to similar bounds. We investigate related bounds and algorithms in a following work (Corah and Michael 2018).

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 $1+\eta$ empirically without needing to obtain the objective value returned by SGA explicitly.

6.3 Worst-case suboptimality

The main focus of this work is to investigate how aspects of problem structure can be used to achieve efficient planning with minimal impact on suboptimality. This contrasts starkly to the worst-case suboptimality which is inversely proportional to the number of robots. While worst-case results by Gharesifard and Smith (2017) and Mirzasoleiman et al. (2013) are relevant, Grimsman et al. (2017) recently proved a somewhat tighter bound that is readily applicable to Algorithm 1.

Theorem 3 (Worst-case suboptimality) *The following worst-case bound for Algorithm 1 holds:*

$$I(M; Y^*) \leq (1 + \eta\lceil n_r/n_d \rceil) I(M; Y^d). \tag{21}$$

The proof can be found in Appendix B.

According to this bound, the planner may entirely fail to take advantage of additional robots if additional planning rounds are not introduced. However, this does not account for locality which can lead distant robots to become decoupled in the solution. This bound can then be interpreted as a limiting scenario for when locality does not hold such as due to extremely close proximity or complex interactions between observations and the environment model.

6.4 Subset selection strategies

The subset selection step of DSGA itself uses a greedy strategy. Looking at this more directly, the negation of the contribution of a single round is

$$I\left(M; Y^d_{D_i} | Y^d_{F_{i-1}}\right) - \sum_{j=1}^{|D_i|} I\left(M; Y^d_{D_{i,j}} | Y^d_{F_{i-1}}\right), \tag{22}$$

found by application of the chain rule of mutual information to (28). Equation (22) is submodular and non-increasing unlike the non-decreasing objectives considered previously. While we apply a heuristic approach and evaluate results empirically, other works have evaluated this setting more directly (Gharan and Vondrák 2011).

More generally, approaches that seek to minimize (22) over the course of individual rounds may fail for sufficiently unbalanced problems. For example, a large number of robots with zero contribution could cause all robots with non-zero objective values to be selected at once at significant detriment to solution quality. While such scenarios are not encountered in this work, this is an important direction for future work.

6.5 Algorithm run time analysis

We compare the run time of DSGA to SGA for variable numbers of robots with run time 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. The messages have fixed sizes and correspond either to a finite-horizon plan or a difference in mutual information. Given these assumptions, broadcast and reduction steps each require $O(\log n_r)$ time (Ladner and Fischer 1980). Let $\alpha(n_r, b_t)$ bound the run time of the selected single-robot planner, noting the dependence on the number of robots and the environment. Similarly, let $\beta(n_r, b_t)$ bound cost of the mutual information computation in Algorithm 1, line 16.

SGA consists of n_r planning steps, each with one broadcast step. The computation time of sequential greedy assignment is then

$$\text{SGA}: \quad O(n_r \alpha(n_r, b_t) + n_r \log n_r). \tag{23}$$

Each of the n_d rounds in DSGA begins with a single planning phase, coming to a cost of $O(n_d \alpha(n_r, b_t))$. The rest of the algorithm consists of subset selection, broadcast of the chosen plans, and computation of mutual information which cumulatively occur once per robot for a total cost of

$$\text{DSGA}_{n_d}: \quad O(n_d \alpha(n_r, b_t) + n_r \beta(n_r, b_t) + n_r \log n_r). \tag{24}$$

If n_d grows slowly or is constant, the performance of DSGA relative to SGA depends on the relative cost of the mutual information computation (β).

For the approximation developed by Charrow et al. (2015b), evaluation of mutual information is linear in the number of cells of the map that are observed. Given the assumption of bounded sensor range, evaluation of mutual information scales linearly in the number of robots so that $\beta(n_r, b_t) \in O(n_r)$. When planning time is dominated by a constant number of mutual information evaluations, such as when using a Monte-Carlo tree search planner with a fixed number of sample trajectories, then $\alpha(n_r, b_t) \in O(n_r)$ so that both algorithms are quadratic. However, in practice, α includes a large constant factor leading to a significant speedup for DSGA_{n_d} compared to SGA. In general, α may also depend non-trivially on factors such as the length of the plan or the scale or complexity of the environment. This further emphasizes the significance of the of eliminating the n_r coefficient from the single-robot planning step.

7 Persistent safe exploration given vehicle dynamics

Up to this point, the analysis has considered maximization of the mutual information objective over a joint space of finite-horizon trajectories which has the form of a partition matroid constraint. However, inter-robot collision constraints cannot be modeled using a matroid, and further challenges arise if the planner can fail to provide a solution or if no solutions exist.

7.1 Independence systems, collision constraints, and suboptimality

After including inter-robot collision constraints, the space of feasible joint plans no longer has the structure of a partition matroid or even a general matroid. Consider, a trajectory that is free of collisions with one assignment of trajectories to a subset of robots. This trajectory is not necessarily free of collisions with another assignment of trajectories to the same subset of robots and neither can a robot in the subset be assigned an additional trajectory from the first assignment. This violates the exchange property for matroids (6). Instead, the constraints can be modeled using an independence system as described in Sect. 3.3, given that subsets of collision-free trajectories are also collision-free. Greedy algorithms can perform arbitrarily poorly for general independence systems (Nemhauser et al. 1978b). However, we can recover a lower bound of $1/(\eta n_r)$ by maximizing mutual information using the first selection in each planning round which is otherwise under-determined because the differences in Algorithm 1, line 9 are uniformly zero for the first plan selected in each round. The bound follows by observing that the maximal feasible subsets have cardinality of at most n_r and by applying suboptimality of the single-robot planner. This result will be used in discussion of system liveness in Sect. 7.3.

Although suboptimality bounds change with the constraint structure, this is not necessarily indicative of typical performance in exploration. Mutual information objectives as used in exploration often favor configurations with large inter-robot distances that minimize redundancy in sensor observations. Robots are also often small compared to their sensor footprints which further mitigates detrimental effects of collision constraints.

7.2 Sufficient requirements for safety

Safety is interpreted as meaning that robots do not enter any occupied part of the environment and do not collide with each other, and each condition is implemented using thresholds on distances as discussed in more detail in Sect. 8. Safety is guaranteed by enforcing the invariant that the current joint plan is safe and terminates in a controlled invariant state[3] as is common in model predictive control (Rawlings and Muske 1993). Plans are executed in a receding-horizon fashion so that robots do not necessarily enter the planned invariant states whereas in the case of planner failure, one or more robots may enter invariant states and remain in those states until a feasible plan can be found.

Safety will be guaranteed under the following assumptions: *1) the initial state, at the start of the exploration process, is an invariant state*; and *2) states that are believed not to be in collision with the environment are free of collisions with the environment and remain so for all time*. The first assumption ensures safety by induction. The second prevents arbitrary changes in the conditions for safety. This latter assumption does not always hold in practice. Approaches, such as selectively relaxing constraints, can address this if necessary.

Assume that the team of robots is tracking a plan that is safe for all time (i.e. meets the stated requirements for all time). We modify Algorithm 1 to only commit to a single-robot plan if the resulting joint plan that results from swapping the old single-robot plan with the new one is also safe for all time. The following constraints are also implemented in the single-robot planner: *1) plans terminate in an invariant state*; and *2) plans respect collision constraints with other robots and the environment according to the full and current joint plan at planning time*. The joint plan which results from replacing a robot's plan with the output of the single-robot planner may not be safe either due to failure to meet the above constraints or due to updates to other robots' plans. In this case, the system falls back to the robot's prior plan which is known to be safe. As a result, the joint plan is always safe for all time.

7.3 Liveness in multi-robot exploration

Informally, liveness properties refer to guarantees that a system will make progress toward a goal. In exploration, a natural statement of liveness for some system state is that the multi-robot team will eventually select an action that reduces the entropy of the map. Although liveness is not guaranteed in general, the proposed system design does admit a limited liveness guarantee. This prevents scenarios such as when a small number of conflicting robots can create a persistent state of deadlock in the entire system.

Assume that some robot finds a feasible single-robot plan with a mutual information reward of at least ϵ. Then, using the results and modifications described in Sect. 7.1, the result-

[3] In the implementation, robots are required to come to a stop and trajectories are then temporally extended as necessary. Although these conditions can be relaxed, such as to invariant sets, there do not appear to be clear or significant benefits from doing so.

ing joint plan also provides a mutual information reward of at east ϵ which in turn corresponds to ϵ entropy reduction in expectation. This condition guarantees liveness in exploration with high probability (i.e. barring infinite sequences of disinformative observations) so long as some robot can find an action that reduces the entropy of the map.

8 Results and discussion

The results are divided into two main parts. Section 8.1 does not incorporate inter-robot collisions and addresses scalability and performance of Algorithm 1 as well as the suboptimality in terms of Theorem 2 in the intended context of submodular maximization over a partition matroid. An earlier version of these results appeared in (Corah and Michael 2017). Section 8.2 introduces inter-robot collisions and the application to a team of aerial vehicles both in simulation and on real hardware. These experiments incorporate the approach described in Sect. 7 in order to address additional challenges related to dynamics and collision prevention.

8.1 Exploration with large numbers of kinematic quadrotors

The proposed approach is first evaluated in a simplified scenario with a team of kinematic quadrotors, disregarding dynamics and inter-robot collisions and so the details discuss in Sect. 7 do not come into play. Performance is evaluated in terms of planner objective values, computation time, and entropy reduction with respect to the map.

8.1.1 Exploration scenario

We test the exploration methodology in a confined and cluttered environment with obstacles (cubes) of various sizes and with robot positions initialized randomly near a lower edge as depicted in Fig. 1. The environment is bounded by a 6 m $\times$ 6 m $\times$ 6 m cube, and the robots model this environment using a 3D occupancy grid with a 0.1 m resolution. The confines and clutter ensure that robots remain proximal to each other. This leads to redundant observations and potential for suboptimal joint plans over the planning horizon. The experiments compare DSGA_1 through DSGA_3 (noting that DSGA_{n_d} represents Algorithm 1 with n_d planning rounds) to SGA for 4, 8, 16, and 32 robots over 3000 iterations (time-steps). Tests evaluating exploration performance are each run twenty times each with randomized initializations. Experiments that evaluate computation time use specially-instrumented planner and single trial for each configuration.

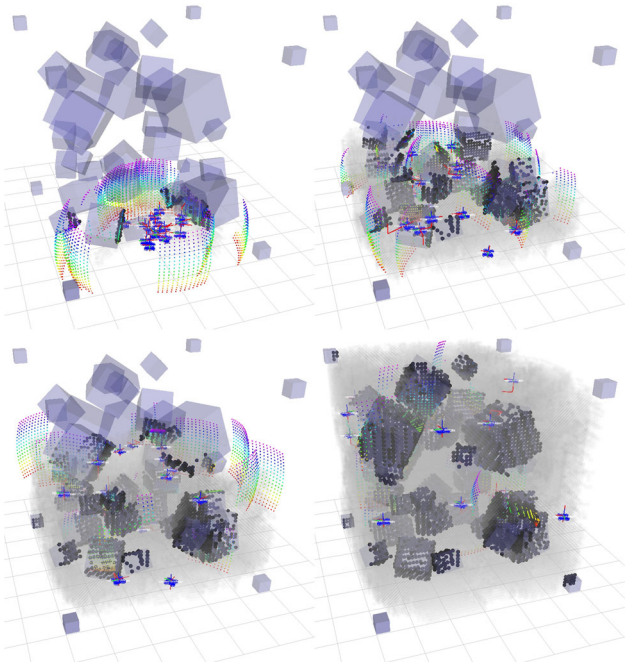


Fig. 1 An example exploration experiment. A multi-robot team explores a three-dimensional environment cluttered with numerous obstacles (cubes) while using an online distributed planner. Known empty space is gray, and occupied space is black. Robots are shown in blue with red trajectories and obtain 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 (Color figure online)

8.1.2 Implementation details

We evaluate the proposed algorithm in simulation and run experiments on a computer equipped with an Intel Xeon E5-2450 CPU (2.5 GHz). The planner and other components of the system are implemented using C++ and ROS (Quigley et al. 2009). When timing the distributed planner, planning steps that would normally be performed in parallel (single-robot planning during each planning round and information propagation) are executed serially. The duration of each such step is taken as the maximum duration of the serially computed steps in order to mimic distributed computation. In practice, computing the reduction to find the minimum excess term (Algorithm 1, line 9) requires an insignificant amount of time. So, although the analysis assumes a logarithmic-time parallel, we compute this by iteration over all elements in the implementation.

The simulated robots emulate kinematic quadrotors moving in a three-dimensional environment with planning, mapping, and mutual information computation each performed in 3D. The single-robot Monte-Carlo tree search planner is

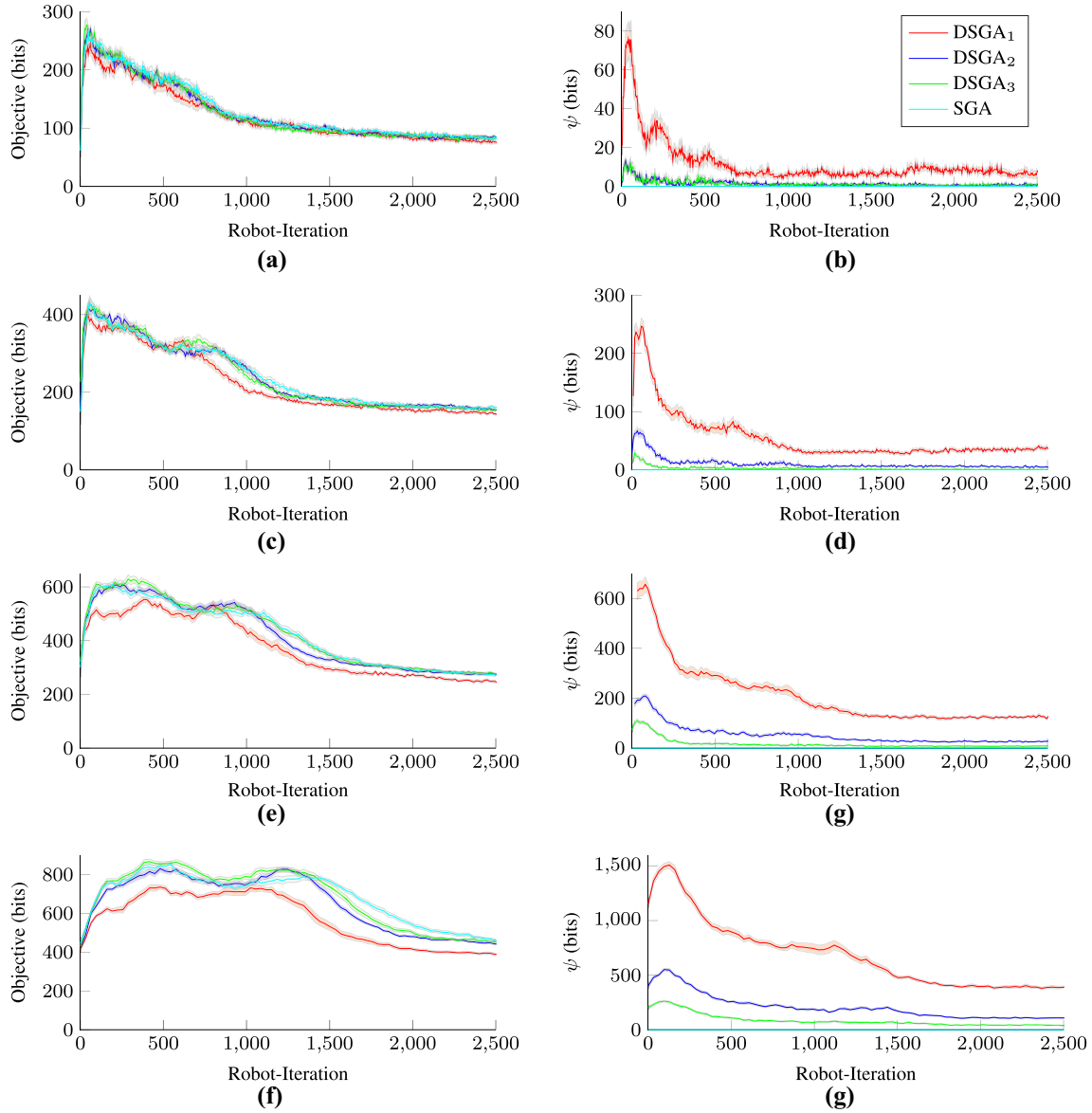


Fig. 2 Objective and ψ values for varying numbers of robots and distributed planning rounds: (left) Mutual information objective values for DSGA$_2$ and DSGA$_3$ closely track SGA while the performance of DSGA$_1$ (which ignores inter-robot interactions) degrades with increasing numbers of robots. (right) Variations in the excess, ψ, terms correspond with the objective values—although ψ-values increase with the numbers of robots, they also decrease with increasing numbers of planning rounds and remain small compared to the objective for DSGA$_3$. Transparent patches indicate standard-error. **a** Information reward, $n_r = 4$, **b** ψ terms, $n_r = 4$, **c** Information reward, $n_r = 8$, **d** ψ terms, $n_r = 8$, **e** Information reward, $n_r = 16$, **f** ψ terms, $n_r = 16$, **g** Information reward, $n_r = 32$, **h** ψ terms, $n_r = 32$ (Color figure online)

run for a fixed number of samples (200) using a discrete set of actions consisting of the choice of translations of ± 0.3 m in the x–y–z directions or heading changes of $\pm \pi/2$ rad. Each robot is equipped with a simulated time-of-flight camera with a range of 2.4 m which is similar to typical depth cameras and having a 19×12 resolution and a $43.6° \times 34.6°$ field-of-view which is oriented with the long axis aligned vertically to promote effective sweeping motions. For efficient evaluation of the mutual information objective, we substitute the approximation of Cauchy-Schwarz mutual information described by Charrow (2015) for the Shannon mutual information (3) and downsample rays by two in each direction. Rather than the uniform prior for cell occupancy (50%) which is commonly used in mapping, we introduce a prior of 12.5% probability of occupancy during evaluation of mutual information (Tabib et al. 2016). This encourages selection of actions that observe larger volumes of unknown space. This set of experiments is not run in real-time although the next is. The planner main-

Table 1 Exploration performance results for the kinematic scenario: The reduction rate is computed with respect to the map over the first 250 robot-iterations and is representative of nominal performance. The total entropy reduction shows the entropy reduction with respect to the map at the end of the experimental trial (3000 robot-iterations)

Algorithm	n_r	Objective $\left(\frac{\text{bits}}{\text{robot}}\right)$		ψ $\left(\frac{\text{bits}}{\text{robot}}\right)$		Entropy red. rate $\left(\frac{\text{bits}}{\text{robot-iter}}\right)$		Total entropy red. (bits)	
		Avg.	SD	Avg.	SD	Avg.	SD	Avg.	SD
$DSGA_1$	4	2.84e+01	1.29e+01	2.76e+00	3.90e+00	2.28e+02	4.83e+01	1.54e+05	1.21e+04
$DSGA_2$	4	2.99e+01	1.30e+01	3.39e-01	8.50e-01	2.36e+02	2.96e+01	1.61e+05	2.04e+03
$DSGA_3$	4	2.99e+01	1.34e+01	3.15e-01	8.75e-01	2.32e+02	2.31e+01	1.61e+05	2.41e+03
SGA	4	3.06e+01	1.34e+01	0.00e+00	0.00e+00	2.30e+02	3.68e+01	1.60e+05	2.85e+03
$DSGA_1$	8	2.63e+01	1.04e+01	6.49e+00	5.80e+00	2.07e+02	1.50e+01	1.60e+05	2.23e+03
$DSGA_2$	8	2.82e+01	1.08e+01	1.23e+00	1.66e+00	2.10e+02	1.78e+01	1.63e+05	1.89e+03
$DSGA_3$	8	2.81e+01	1.10e+01	2.67e-01	6.30e-01	2.04e+02	1.41e+01	1.64e+05	1.75e+03
SGA	8	2.84e+01	1.06e+01	0.00e+00	0.00e+00	2.05e+02	1.53e+01	1.64e+05	2.80e+03
$DSGA_1$	16	2.22e+01	7.47e+00	1.24e+01	8.14e+00	1.64e+02	1.11e+01	1.60e+05	1.77e+03
$DSGA_2$	16	2.46e+01	8.08e+00	3.04e+00	2.58e+00	1.77e+02	9.52e+00	1.66e+05	1.42e+03
$DSGA_3$	16	2.52e+01	8.17e+00	9.67e-01	1.30e+00	1.78e+02	8.97e+00	1.67e+05	1.67e+03
SGA	16	2.48e+01	7.86e+00	0.00e+00	0.00e+00	1.74e+02	7.60e+00	1.68e+05	1.65e+03
$DSGA_1$	32	1.69e+01	4.64e+00	2.00e+01	1.00e+01	1.26e+02	4.60e+00	1.61e+05	1.75e+03
$DSGA_2$	32	1.94e+01	4.99e+00	6.00e+00	3.59e+00	1.27e+02	5.45e+00	1.67e+05	1.44e+03
$DSGA_3$	32	2.02e+01	5.30e+00	2.44e+00	1.79e+00	1.29e+02	7.60e+00	1.69e+05	9.94e+02
SGA	32	2.03e+01	4.63e+00	0.00e+00	0.00e+00	1.31e+02	8.73e+00	1.72e+05	6.51e+02

tains the robot states internally and triggers the camera after each iteration.

8.1.3 Evaluation of planner performance and objective values

Figure 2 and Table 1 show results for exploration experiments comparing $DSGA_1$ through $DSGA_3$ to SGA. The excess (ψ) is largest at the beginning of each exploration run as all robots start near the same location. As the robots spread out, all planners approach approximately steady-state conditions in terms of both excess suboptimality and objective values before values decay once the environment is mostly explored. The ψ terms remain relatively large for $DSGA_1$—which assigns all plans in a single round and does not reason about conditional dependencies. However, the ψ terms decrease monotonically with increasing numbers of planning rounds. These values remain small for $DSGA_3$ and are approximately one-eighth of the objective value on average for 32 robots whereas values for $DSGA_1$ exceed the value of the objective for the same number of robots. Decreasing values of ψ are reflected in plots of the mutual information objective. Whereas $DSGA_2$ and $DSGA_3$ closely track SGA, objective values for $DSGA_1$ degrade with increasing numbers of robots.

When interpreting these results and entropy reduction results to be shown in Sect. 8.1.5, it is helpful to note that the scale of the environment remains constant and that increasing the number of robots serves to increase a notion of the density of the distribution of robots in the environment. In this sense, similar performance can be expected of larger teams operating in larger environments up to constraints on computation and communication. Toward this end, the next subsection evaluates computation times for DSGA and demonstrates that it scales much better than SGA in practice.

8.1.4 Computational performance

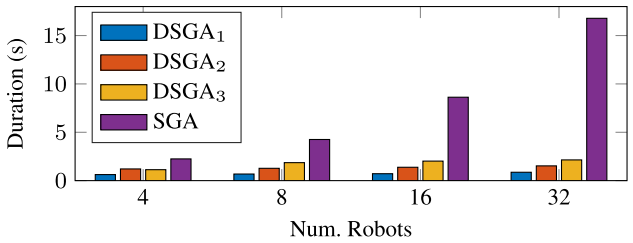
While planning performance is largely consistent, DSGA is able to take advantage of distributed computation to provide significantly improved computation times (Fig. 3). Computation times for SGA scale approximately linearly in our simulation experiments. The average computation time for SGA increases more than seven times from $n_r = 4$ to $n_r = 32$ robots from 2.25 to 16.8 s while times for DSGA remain small and decrease slowly. As a result, $DSGA_3$ provides a 2–8 times speedup compared to SGA. With 32 robots, the computation time increases by only a factor of 2.5 from $DSGA_1$ to $DSGA_3$, despite tripling the number of planning rounds. These results indicate that DSGA is able to provide multi-robot coordination for large numbers of robots while accommodating requirements for online performance well after doing the same with SGA becomes intractable.

8.1.5 Entropy reduction performance

Figure 4 compares DSGA to SGA for various numbers of robots and is summarized in Table 1. Exploration rates early in exploration runs remain consistent across planners and degrade by approximately 43% as the numbers of robots increase due to crowding. This translates to significant improvements in the total rate of exploration even as efficiency decreases.

Alg.	n_r	S.R. Planning (s)		Prop (s)		Total (s)	
		Avg.	SD	Avg.	SD	Avg.	SD
$DSGA_1$	4	0.600	0.0603	0.0251	0.00529	0.625	0.0610
$DSGA_2$	4	1.19	0.0928	0.0181	0.00422	1.21	0.0934
$DSGA_3$	4	1.11	0.106	0.0189	0.00415	1.13	0.107
SGA	4	2.23	0.150	0.0184	0.00253	2.25	0.151
$DSGA_1$	8	0.630	0.0562	0.0524	0.00664	0.682	0.0577
$DSGA_2$	8	1.22	0.0863	0.0491	0.00696	1.27	0.0886
$DSGA_3$	8	1.82	0.105	0.0428	0.00708	1.86	0.108
SGA	8	4.22	0.243	0.0342	0.00381	4.25	0.245
$DSGA_1$	16	0.609	0.0565	0.106	0.00916	0.715	0.0616
$DSGA_2$	16	1.28	0.0867	0.106	0.00954	1.38	0.0912
$DSGA_3$	16	1.91	0.0911	0.106	0.00985	2.02	0.0923
SGA	16	8.55	0.329	0.0667	0.00543	8.63	0.332
$DSGA_1$	32	0.649	0.0565	0.217	0.0105	0.865	0.0596
$DSGA_2$	32	1.30	0.0547	0.222	0.0110	1.53	0.0583
$DSGA_3$	32	1.92	0.0949	0.218	0.0124	2.14	0.101
SGA	32	16.6	0.592	0.131	0.00816	16.8	0.596

(a)



(b)

Fig. 3 Computational performance (seconds) in terms of total computation time (time elapsed from when the first robot starts planning until the last robot stops). **a** Table of timing data: 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 DSGA (Color figure online)

The entropy reduction performance of $DSGA_2$ and $DSGA_3$ closely matches SGA for each number of robots. Interestingly, the performance of $DSGA_1$ is worst for $n_r = 4$ robots where the rate of entropy reduction degrades after initially performing well. For larger numbers of robots, differences in entropy reduction performance are most apparent in the total entropy reduction. This is likely a result of the planners involving inter-robot coordination being able to distribute robots across the environment more effectively. Because we use local finite-horizon planner, $DSGA_1$ then occasionally fails to allocate robots to and explore some portions of the environment. Given a spatially global planning strategy, longer exploration times could then be expected for $DSGA_1$ due to the cost of traveling to these unexplored regions of the environment.

8.2 Simulation and experiments of dynamic robots with inter-robot collision constraints

This section addresses more realistic systems and involves inter-robot collision avoidance and non-trivial dynamics in the form of teams of real and simulated quadrotors. Unlike the previous section, robots now track dynamic trajectories while planning is performed in real-time. Simulation results for a team of six robots demonstrate that DSGA retains performance comparable to SGA. Experiments using a three real quadrotors serve to ground the simulation results by demonstrating comparable results on a similarly configured system in practice.

8.2.1 Implementation details

As these experiments now involve inter-robot collision constraints, we now incorporate the details Sect. 7 into both the SGA and DSGA planners. Rather than discrete steps, the Monte-Carlo tree search planner now uses a library of polynomial motion primitives similar to that described by Tabib et al. (2016) to provide actions. Because trajectories are executed in real-time, the Monte-Carlo tree search planner is run in a fully any-time fashion at a rate of 1/3 Hz rather than for a fixed number of samples, and time in single-robot planning is budgeted according to the number of sequential planning steps in the given multi-robot planner. The planner uses a finite 6 s horizon with motion primitives whose durations vary from 0.5 to 10 s (2 s is typical), and at each planning step as many as 162 motion-primitive actions are available although some may be infeasible due to collision constraints. Robot-environment collision checks are implemented use a truncated distance field for efficient lookups, and inter-robot collision checks use horizontal distances to prevent any robot from flying in another's downwash. This set of experiments uses a custom simulation and control framework that implements a standard quadrotor dynamics model and a quadrotor controller that tracks polynomial trajectories in the differentially-flat outputs (Mahony et al. 2012) and a team of custom quadrotors equipped with Structure brand depth sensors.

8.2.2 Simulation results

The simulation trials (Fig. 5) evaluate a team of six robots exploring a pseudo-planar rearrangement of the environment shown in Figure 1 using sequential planning (SGA), myopic planning ($DSGA_1$), and distributed planning with three rounds ($DSGA_3$) with twenty trials per each planner. Single-robot planning steps (Monte-Carlo tree search) were run in parallel on the same computer as in Sect. 8.1.2.

8.2.3 Experimental results

The simulation environment and results for entropy reduction are evident shown in Fig. 5. The exploration performance is largely similar across planners except that $DSGA_1$ exhibits a slight delay at the start due to a tendency for robots to

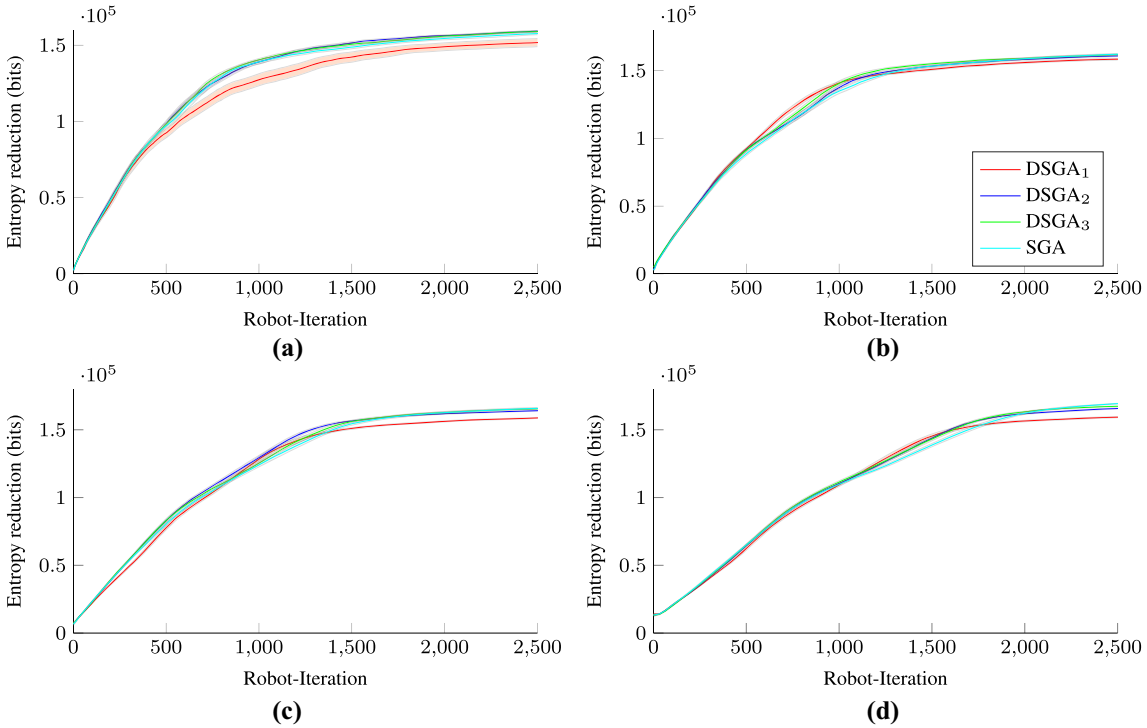


Fig. 4 Entropy reduction performance with different numbers of robots and planner configurations: $DSGA_2$ and $DSGA_3$ closely track SGA in all cases while $DSGA_1$, which does not incorporate inter-robot coordination, performs worst for $n_r = 4$ and has reduced total entropy reduction for $n_r = 16$ and $n_r = 32$. Transparent patches show standard-error. **a** Information gain per robot ($n_r = 4$). **b** Information gain per robot ($n_r = 8$). **c** Information gain per robot ($n_r = 16$). **d** Information gain per robot ($n_r = 32$) (Color figure online)

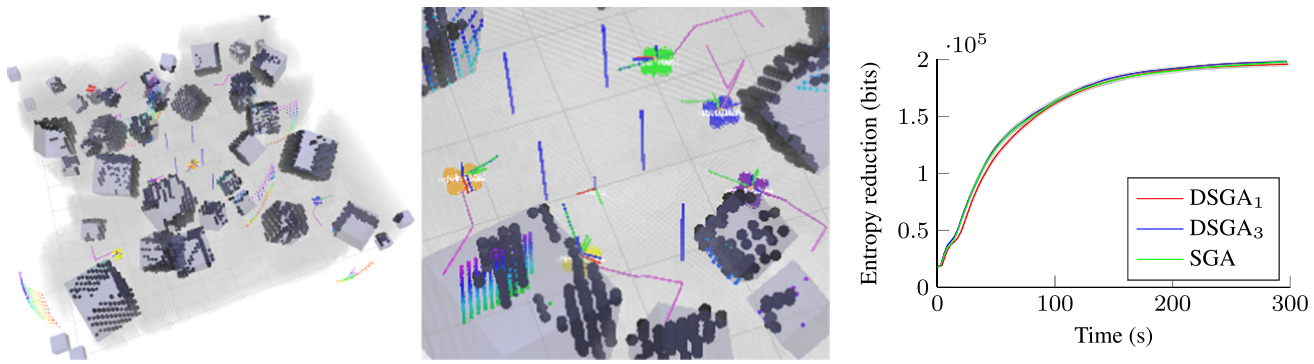


Fig. 5 (left) Six robots explore a pseudo-planar environment using motion-primitive trajectories. (middle) Robots (such as the green and blue) plan informative and collision-free trajectories using $DSGA_3$ despite operating in close proximity. The portion of each robot's receding-horizon plan that is currently being executed is colored in green, and the rest of the horizon is magenta. (right) Planners perform similarly in terms of entropy reduction although the initial performance of $DSGA_1$ is impaired as it frequently resorts to using fallback trajectories due to conflicts in planned trajectories (Color figure online)

choose conflicting trajectories. Because robots are initialized near each other, toward the center of the environment, (with blue lines marking the locations where robots took off) the single-robot planner returns plans that would result in inter-robot collisions most often early in each trial. This is evident in Fig. 6 which plots the cumulative number of times that the multi-robot planner has had to fallback to the existing safe trajectory due to either inter-robot collisions or failure of single-robot planning as described in Sect. 7.2. This is most pronounced for $DSGA_1$ due to the lack of coordination in the planner. Using the fallback trajectories is most detrimental at the very beginning of the trial when the fallback consists of staying stationary at the starting position, causing the delay. Later, once they have been populated with prior

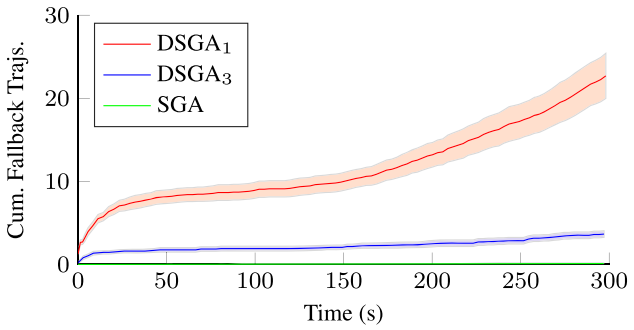


Fig. 6 Cumulative numbers of fallback trajectories selected by multi-robot planner in simulation results with dynamic quadrotors: The multi-robot planners resort to fallback trajectories when the single-robot planner fails or returns a plan that would result in inter-robot collisions. $DSGA_1$ uses fallback trajectories frequently early in trials when robots are near each other and later after most of the environment has been explored (Color figure online)

planning results, resorting to fallback trajectories becomes less significant.

An experimental trial with a team of three quadrotors is also included as depicted in Fig. 7. This experiment serves to ground the simulation results described in the prior subsections on a real system. Due to the small size of the team, only SGA is used for coordination as DSGA would be expected to provide little benefit ($DSGA_3$ is essentially equivalent to SGA for three robots). Planning is performed offboard on a laptop with a Intel i7-5600U CPU (2.6 GHz). Otherwise, the system configuration is identical to the prior set of simulation results. Whereas the simulation results demonstrate application of the approach described in Sect. 7 for exploration with collision constraints and dynamic robots using a significant number of robots, these hardware results connect the implementation and its configuration to a real system.

The experiment is performed in a motion capture flight arena populated with a pile of foam blocks of various shapes and sizes. During the course of the exploration trial, the team of robots is able to successfully navigate and map the environment as evident in the trajectories and cumulative entropy reduction. The rate of entropy reduction per robot at the beginning of the trial is similar to Fig. 7 at approximately $400 \frac{\text{bits}}{\text{s·robot}}$. While the simulation results demonstrate efficiency of SGA and DSGA with and without collision constraints these experimental results demonstrate that the planner design and configuration is also applicable to exploration with a real multi-robot team and performs similarly.

9 Conclusions and future work

The proposed distributed algorithm (DSGA) efficiently approximates sequential greedy assignment (SGA) and is appropriate for implementation on large multi-robot teams using distributed computation and online planning. We have applied this algorithm to the problem of multi-robot exploration, and demonstrated consistent entropy reduction performance in simulation for large numbers of robots exploring and mapping a complex three-dimensional environment. The results demonstrate that DSGA is able to effectively take advantage of distributed computation without degradation in solution quality at scales where application of SGA becomes intractable. We expect that this result will be instrumental in development of physical multi-robot systems that take advantage of distributed computation for exploration and other online informative planning problems.

In order to address more realistic scenarios, we incorporated robot dynamics and inter-robot collision constraints into the problem formulation. Although collision constraints result in increased problem complexity, we are able to modify the planning approach to guarantee safety and liveness under weak conditions. Simulation results demonstrate consistent exploration performance and indicate that collision constraints have a relatively minor impact on exploration performance. Further, including collision constraints makes this approach viable for application to real robots, and we demonstrate this on a team of aerial robots flying in motion capture.

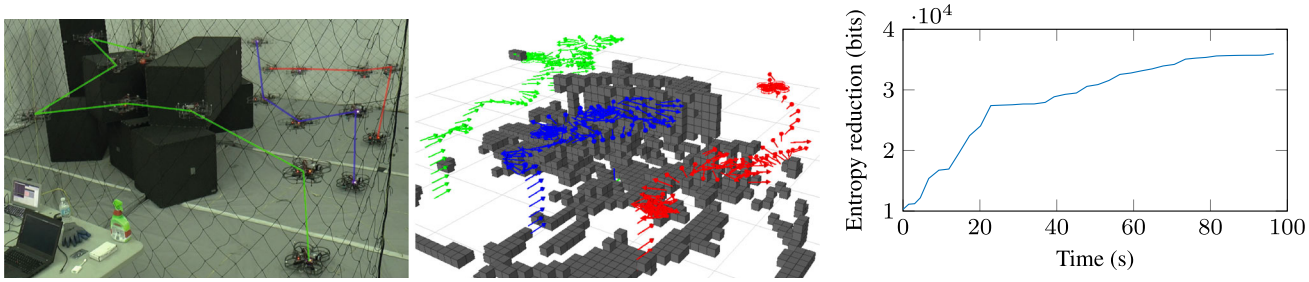


Fig. 7 (left) A team of three robots explores a motion capture arena occupied by a geometric object consisting of foam boxes using Monte-Carlo tree search and motion-primitive trajectories and coordination via SGA and (middle) produce a voxel grid map of the environment (shown from a different vantage-point to also detail the robot in red which explores behind the obstacle). (right) The entropy of the map decreases as the robots move through the environment (Color figure online)

Although planning in DSGA is not entirely sequential, the assignment in the subset selection step is, and the resulting asymptotic computation time is identical to sequential greedy assignment in the context of this work despite providing significant improvements in practice. Further analysis, such as using the distribution of robots and information, could be useful for development of approaches with improved asymptotic performance and to provide stronger performance bounds.

10 Appendix A: Proof of Theorem 2

Proof The proof of the suboptimality bound relating DSGA to SGA incorporates suboptimality of the single-robot planner and is similar to (Singh et al. 2009) or (Atanasov et al. 2015). We obtain the following by monotonicity and by rearranging the resulting telescoping sum

$$\begin{aligned} I(M;Y^*) &\leq I(M;Y^*) + \sum_{i=1}^{n_r} I\left(M; Y_i^d | Y_{1:i-1}^d, Y_{i+1:n_r}^*\right) \\ &= I(M;Y^d) + \sum_{i=1}^{n_r} I\left(M; Y_i^* | Y_{1:i-1}^d, Y_{i+1:n_r}^*\right). \end{aligned} \tag{25}$$

By submodularity

$$I\left(M; Y_i^* | Y_{1:i-1}^d, Y_{i+1:n_r}^*\right) \leq I\left(M; Y_i^* | Y_{1:i-1}^d\right).$$

Without loss of generality, assume that agent indices correspond to the selection order and rewrite in terms of the planning rounds such that $I(M; Y_i^* | Y_{1:i-1}^d) = I(M; Y_{D_{j,k}}^* | Y_{D_{j,1:k-1} \cup F_{j-1}}^d)$ and note that that although Y^* is formally a set, the mapping from elements to robots can be obtained by the intersections $Y^* \cap \mathcal{Y}_i$ given that the sets $\mathcal{Y}_i$ are disjoint. Then, by submodularity,

$$I(M; Y_i^* | Y_{1:i-1}^d) \leq I\left(M; Y_{D_{j,k}}^* | Y_{F_{j-1}}^d\right).$$

By (17) and the greedy maximization step in Algorithm 1

$$I\left(M; Y_{D_{i,j}}^* | Y_{F_{i-1}}^d\right) \leq \eta I\left(M; Y_{D_{i,j}}^d | Y_{F_{i-1}}^d\right). \tag{26}$$

Substitute (26) and preceding inequalities into (25) to obtain

$$I(M;Y^*) \leq I(M;Y^d) + \eta \sum_{i=1}^{n_d} \sum_{j=1}^{|D_i|} I\left(M; Y_{D_{i,j}}^d | Y_{F_{i-1}}^d\right). \tag{27}$$

The distributed objective can be rewritten as a sum so that $I(M;Y^d) = \sum_{i=1}^{n_d} \sum_{j=1}^{|D_i|} I(M; Y_{D_{i,j}}^d | Y_{D_{i,1:j-1} \cup F_{i-1}}^d)$ and substituted into (27) to obtain

$$\begin{aligned} I(M;Y^*) \leq (1+\eta) I(M;Y^d) + \eta \sum_{i=1}^{n_d} \sum_{j=1}^{|D_i|} \Big(& I\left(M; Y_{D_{i,j}}^d | Y_{F_{i-1}}^d\right) \\ & - I\left(M; Y_{D_{i,j}}^d | Y_{D_{i,1:j-1} \cup F_{i-1}}^d\right)\Big) \end{aligned} \tag{28}$$

which expresses the suboptimality in terms of decreases in the conditional mutual information from when results are obtained from the planner to when they are assigned. By rewriting mutual information in terms of entropies we can rearrange to obtain the following

$$I(M;Y_1) - I(M;Y_1|Y_2) = I(Y_1;Y_2) - I(Y_1;Y_2|M).$$

If Y_1 and Y_2 are conditionally independent given M, then the mutual information, $I(Y_1;Y_2|M) = 0$ and so $I(M;Y_1) - I(M;Y_1|Y_2) = I(Y_1;Y_2)$. By substitution into (28) we can obtain the slightly more concise and final expression for the suboptimality in terms of the mutual information between observations

$$\begin{aligned} I(M;Y^*) \leq (1+\eta) I(M;Y^d) \\ + \eta \sum_{i=1}^{n_d} \sum_{j=1}^{|D_i|} I(Y_{D_{i,j}}^d; Y_{D_{i,1:j-1}}^d | Y_{F_{i-1}}^d). \end{aligned} \tag{29}$$

□

11 Appendix B: Proof of Theorem 3

Proof Equation (21) follows from Theorem 1 by Grimsman et al. (2017) which proves a $k+1$ bound where k is the clique cover number of a directed acyclic graph associated with the planner structure. In this directed graph, each robot represents a vertex, and the graph has a directed edge (a, b) between robots $a, b \in \mathcal{R}$ if b maximizes its objective (15) conditional on the sequence of observations selected by a. Here, a clique, which is a complete subgraph, is analogous to a set of robots that plan sequentially given the choices by all prior robots in the clique. For Algorithm 1, any set of robots $A \subseteq \mathcal{R}$ with at most one robot from each planning round ($|A \cap D_i| \leq 1$ for $i \in \{1, \ldots, n_d\}$) forms a clique in the associated directed graph. A clique cover of size $\lceil n_r / n_d \rceil = \max_{i \in \{1,\ldots,n_d\}} |D_i|$ can be obtained by selecting cliques with a single robot (as available) from each planning round $(D_1, \ldots, D_{n_d})$ without replacement. Then, (21) follows by substitution of (17) to obtain a factor of η. □

References

Atanasov, N.A., Le Ny, J., Daniilidis, K., & Pappas, G.J. (2015). Decentralized active information acquisition: Theory and application to multi-robot SLAM. In *Proceedings of the IEEE international conference on robotics and automation, Seattle, WA*.

Barbosa, RdP., Ene, A., Nguyen, H.L., & Ward, J. (2016). A new framework for distributed submodular maximization. In *Proceedings of the IEEE annual symposium on foundations of computer science, New Brunswick, NJ*.

Best, G., Cliff, O. M., Patten, T., Mettu, R. R., & Fitch, R. (2016). Decentralised Monte Carlo tree search for active perception. In *Algorithmic foundation robotics*. San Francisco, CA

Browne, C., Powley, E., Whitehouse, D., Lucas, S., Cowling, P. I., Rohlfshagen, P., et al. (2012). A survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, *4*(1), 1–43.

Calinescu, G., Chekuri, C., Pal, M., & Vondrák, J. (2011). Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, *40*(6), 1740–1766.

Charrow, B. (2015). *Information-theoretic active perception for multi-robot teams*. Ph.D. thesis, University of Pennsylvania.

Charrow, B., Kumar, V., & Michael, N. (2014). Approximate representations for multi-robot control policies that maximize mutual information. *Autonomous Robots*, *37*(4), 383–400.

Charrow, B., Kahn, G., Patil, S., Liu, S., Goldberg, K., Abbeel, P., Michael, N., & Kumar, V. (2015a). Information-theoretic planning with trajectory optimization for dense 3D mapping. In *Proceedings of robotics: science and systems, Rome, Italy*.

Charrow, B., Liu, S., Kumar, V., & Michael, N. (2015b). Information-theoretic mapping using Cauchy-Schwarz quadratic mutual information. In *Proceedings 1990 IEEE international conference on robotics and automation, Seattle, WA*.

Chaslot, G. (2010). *Monte-Carlo tree search*. Ph.D. thesis, Universiteit Maastricht.

Chekuri, C., & Martin, P. (2005). A recursive greedy algorithm for walks in directed graphs. In *Proceedings of the IEEE annual symposium on foundations of computer science*, pp 245–253.

Choi, H. L., Brunet, L., & How, J. P. (2009). Consensus-based decentralized auctions for robust task allocation. *IEEE Transactions on Robotics*, *25*(4), 912–926.

Corah, M., & Michael, N. (2017). Efficient online multi-robot exploration via distributed sequential greedy assignment. In *Proceedings of robotics: science and system, Cambridge, MA*.

Corah, M., & Michael, N. (2018). Distributed submodular maximization on partition matroids for planning on large sensor networks. In *Proceedings of the IEEE conference on decision and control. Miami, FL* (**submitted for publication**).

Cover, T. M., & Thomas, J. A. (2012). *Elements of information theory*. New York, NY: Wiley.

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *IEEE Computer Society*, *22*(6), 46–57.

Filmus, Y., & Ward, J. (2012). A tight combinatorial algorithm for submodular maximization subject to a matroid constraint. In *Proceedings of the IEEE annual symposium on foundations of computer science, New Brunswick, NJ*.

Gharan, S.O., & Vondrák, J. (2011). Submodular maximization by simulated annealing. In *Proceedings of the symposium on discrete algorithms, Philadelphia, PA*.

Gharesifard, B., & Smith, S.L. (2017). Distributed submodular maximization with limited information. *IEEE Transactions on Control of Network Systems*. https://doi.org/10.1109/TCNS.2017.2740625.

Goundan, P. R., & Schulz, A. S. (2007). Revisiting the greedy approach to submodular set function maximization. *Optim Online*, *1984*, 1–25.

Grimsman, D., Ali, M.S., Hespanha, P., & Marden, J.R. (2017). Impact of information in greedy submodular maximization. In *Proceedings of the IEEE conference on decision and control, Melbourne, Australia*.

Jadidi, M.G., Miro, J.V., & Dissanayake, G. (2015). Mutual information-based exploration on continuous occupancy maps. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, Hamburg, Germany*.

Jorgensen, S., Chen, R.H., Milam, M.B., & Pavone, M. (2017). The matroid team surviving orienteers problem: Constrained routing of heterogeneous teams with risky traversal. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, Vancouver, Canada*.

Julian, B. J., Karaman, S., & Rus, D. (2014). On mutual information-based control of range sensing robots for mapping applications. *The International Journal of Robotics Research*, *33*(10), 1357–1392.

Krause, A., & Guestrin, C.E. (2005). Near-optimal nonmyopic value of information in graphical models. In *Proceedings of the conference on uncertainty in artificial intelligence, Edinburgh, Scotland*.

Krause, A., Singh, A., & Guestrin, C. (2008). Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, *9*, 235–284.

Ladner, R. E., & Fischer, M. J. (1980). Parallel prefix computation. *Journal of the ACM (JACM)*, *27*(4), 831–838.

Lauri, M., & Ritala, R. (2016). Planning for robotic exploration based on forward simulation. *Robotics and Autonomous Systems*, *83*, 15–31.

Mahony, R., Kumar, V., & Corke, P. (2012). Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics and Automation Magazine*, *19*(3), 20–32.

Mirzasoleiman, B., Karbasi, A., Sarkar, R., & Krause, A. (2013). Distributed submodular maximization: Identifying representative elements in massive data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani & K. Q. Weinberger (Eds.), *Procdeedings of the advances in neural information processing systems* (Vol. 26, pp. 2049–2057). Stateline, Nevada: Curran Associates, Inc. http://papers.nips.cc/paper/5039-distributed-submodular-maximization-identifying-representative-elements-in-massivedata.pdf.

Nelson, E., & Michael, N. (2015). Information-theoretic occupancy grid compression for high-speed information-based exploration. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, Hamburg, Germany*.

Nemhauser, G. L., & Wolsey, L. A. (1978). Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, *3*(3), 177–188.

Nemhauser, G. L., Wolsey, L. A., & Fisher, M. L. (1978a). An analysis of approximations for maximizing submodular set functions-I. *Mathematics Program*, *14*(1), 265–294.

Nemhauser, G. L., Wolsey, L. A., & Fisher, M. L. (1978b). An analysis of approximations for maximizing submodular set functions-II. *Polyhedral Combinatorics*, *8*, 73–87.

Patten, T. (2017). *Active object classification from 3D range data with mobile robots*. Ph.D. thesis, The University of Sydney.

Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., et al. (2009). ROS: An open-source robot operating system. In *ICRA workshop on open source software, Kobe, Japan*

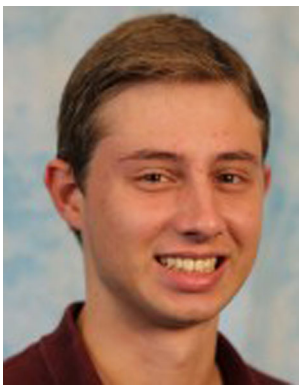
Rawlings, J. B., & Muske, K. R. (1993). The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, *38*(10), 1512–1516.

Regev, T., & Indelman, V. (2016). Multi-robot decentralized belief space planning in unknown environments via efficient re-evaluation of impacted paths. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, Daejeon, Korea*.

Singh, A., Krause, A., Guestrin, C., & Kaiser, W. J. (2009). Efficient informative sensing using multiple robots. *Journal of Artificial Intelligence Research*, *34*, 707–755.

Tabib, W., Corah, M., Michael, N., & Whittaker, R. (2016). Computationally efficient information-theoretic exploration of pits and

caves. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, Daejeon, Korea*.
Williams, J.L. (2007). *Information theoretic sensor management*. Ph.D. thesis, Massachusetts Institute of Technology.
Williams, R.K., Gasparri, A., & Ulivi, G. (2017). Decentralized matroid optimization for topology constraints in multi-robot allocation problems. In *Proceedings of the IEEE international conference on robotics and automation, Singapore*.
Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. In *Proceedings of the international symposium on computer intelligence in robotics and automation, Monterey, CA*.
Zhou, T., Ouyang, H., Chang, Y., Bilmes, J., & Guestrin, C. (2017). Scaling submodular maximization via pruned submodularity graphs. *Proceedings of Machine Learning Research*, *54*, 316–324.



Micah Corah is a Ph.D. student in Robotics Institute at Carnegie Mellon University. Micah received B.S. degrees in Computer Science and Mechanical Engineering from the Rensselaer Polytechnic Institute in 2015. His research interests are in informative-planning and exploration for single and multi-robot systems.

Nathan Michael is an Associate Research Professor in the Robotics Institute at Carnegie Mellon University in Pittsburgh, Pennsylvania. His research focuses on enabling autonomous ground and aerial vehicles to robustly operate in uncertain environments with emphasis on robust and adaptive perception, control, and cooperative autonomy.