

Scalable Distributed Planning for Multi-Robot, Multi-Target Tracking

Micah Corah and Nathan Michael

Abstract—In multi-robot multi-target tracking, robots coordinate to monitor groups of targets moving about an environment. We approach planning for such scenarios by formulating a receding-horizon, multi-robot sensing problem with a mutual information objective. Such problems are NP-Hard in general. Yet, our objective is submodular which enables certain greedy planners to guarantee constant-factor suboptimality. However, these greedy planners require robots to plan their actions in sequence, one robot at a time, so planning time is at least proportional to the number of robots. Solving these problems becomes intractable for large teams, even for distributed implementations. Our prior work proposed a distributed planner (RSP) which reduces this number of sequential steps to a constant, even for large numbers of robots, by allowing robots to plan in parallel while ignoring some of each others’ decisions. Although that analysis is not applicable to target tracking, we prove a similar guarantee, that RSP planning approaches performance guarantees for fully sequential planners, by employing a novel bound which takes advantage of the independence of target motions to quantify effective redundancy between robots’ observations and actions. Further, we present analysis that explicitly accounts for features of practical implementations including approximations to the objective and anytime planning. Simulation results—available via open source release—for target tracking with ranging sensors demonstrate that our planners consistently approach the performance of sequential planning (in terms of position uncertainty) given only 2–8 planning steps and for as many as 96 robots with a 24x reduction in the number of sequential steps in planning. Thus, this work makes planning for multi-robot target tracking tractable at much larger scales than before, for practical planners and general tracking problems.

I. INTRODUCTION

In target tracking problems, robots seek to observe a number of discrete targets whose states may evolve in time, such as for surveillance, monitoring wildlife [1], and intercepting rogue UAVs [2]. The robots may track the positions of the targets as well as other features of their states, such as to track an animal’s actions or the pose of an elite athlete. When a large number of such targets are spread over more space than a single robot can cover, deploying more robots can improve their capacity to track the targets but at the expense of more complex planning problems.

Still, even simple target tracking problems can be difficult to model and solve. Noisy range observations can produce multi-modal posterior distributions over target positions which do not have closed-form solutions; and planning and tracking systems often approximate both filter posteriors and

sensing utility [3, 4]. Likewise, realistic environments induce complexity by constraining target motion such as for search on a road network [5] or in an indoor office space [6]. Additionally, deep learning methods for visual object tracking enable systems to track wide varieties of objects, and these objects may have similarly varied states and dynamics [7]. This motivates development of objectives and planners that capture the nuances of these problems. To address this, we provide analysis for general tracking problems where target states are independent of each other and the robots tracking them¹ with a flexible objective (mutual information), and we allow for choice of approximate representations and planners.

Systems for target tracking in multi-robot settings often rely on greedy algorithms [9, 10] which apply to a wide variety of relevant submodular objectives, including the mutual information objectives we study [11, 12]. Additionally, these greedy algorithms can augment general single-robot planners to provide efficient planning and constant-factor performance guarantees in multi-robot domains [13] despite common formulations of these problems being NP-Hard [11, 14].

However, a limitation of greedy algorithms for multi-robot planning—especially in distributed settings—is that robots must make decisions sequentially [15] so that planning time grows with the number of robots, and this growth in planning time can prohibit large teams of robots from reacting promptly to target motions. Advances in distributed algorithms relevant to target tracking begin to address this issue but are currently limited to coverage-like objectives [16, 17]. Here, we extend our analysis for RSP planning [16, 18] to include more general mutual information objectives. Analysis for sequential planners also typically assumes individual robots obtain solutions that are either exact [15, 19] or within a constant factor of optimal [13, 20, 21]. Although such analysis is appropriate for single-robot planners with guarantees on solution quality [13, 20, 22], assuming constant-factor suboptimality is less suited for anytime or sampling-based planners [15, 23, 24] like those we apply in this paper.

A. Contributions

This paper extends methods for distributed planning to target tracking problems, presents analysis that accounts for common approximations, and applies these methods to multi-robot planning in a simulated target tracking scenario.²

1) *Distributed planning for target tracking*: Although sequential planners generally require computation time that is at least proportional to the number of robots, recent works on

Micah Corah is affiliated with the NASA Jet Propulsion Laboratory (JPL), Pasadena, CA, USA. Nathan Michael is affiliated with the Robotics Institute, Carnegie Mellon University (CMU), Pittsburgh, PA, USA. This work was completed while Micah Corah was a student at CMU.

mcorah@jpl.nasa.gov, nmichael@cmu.edu

¹Although problems may be adversarial in other parameters, this excludes pursuit-evasion problems [2, 8] where targets can observe pursuers.

²This work and [16] appear as chapters in the thesis [18].

distributed optimization introduced methods that can reduce the number of sequential steps [25, 26]. Our prior works built on these to develop planners based on Randomized Sequential Partitions³ (RSP) that run in constant numbers of steps, independent of the number of robots [16, 18]. While the prior analysis is only relevant to coverage objectives,⁴ this paper demonstrates that RSP planning is applicable to target tracking with mutual information objectives by providing guarantees on solution quality in terms of a bound on the effective pairwise redundancy between robots' actions. We obtain this bound after decomposing the objective as a sum of submodular functions over each target. This analysis demonstrates that distributed planners running in constant time can guarantee performance approaching that of more inefficient sequential planners (within half of optimal) [19].

2) *Analysis of approximate, anytime planning:* The analysis of RSP also introduces a novel approach to account for common sources of suboptimality in receding-horizon planning (due to *approximation of the objective and suboptimal single-robot planning*). This affirms that methods for submodular maximization are applicable to target tracking in the presence of approximate objective values and anytime planners that may sometimes produce poor results or fail.

3) *Application to a multi-robot multi-target tracking problem:* Finally, we apply the analysis to develop a planner⁵ for multi-robot multi-target tracking with a mutual information objective and demonstrate that a distributed RSP planner running in constant time can guarantee suboptimality approaching that of fully sequential planning. Then, additional simulation results confirm that RSP maintains consistent solution quality for up to 96 robots. This produces a 24× reduction in the number of sequential planning steps and an at least equivalent reduction in planning duration.

II. BACKGROUND

Let us begin by presenting the mathematical background on our approach to multi-robot planning.

A. Set functions and their properties

Functions of sets $g : 2^\Omega \rightarrow \mathbb{R}$ can quantify the utility of sets of control actions, each a subset of a finite collection of possible actions Ω , and we seek to maximize set functions that satisfy the following conditions. First, g is normalized if $g(\emptyset) = 0$. Additionally, objectives in sensing tasks [6, 13, 15, 27] often have useful monotonicity properties when written as set functions. A function is monotonically increasing if

$$g(A) \geq g(B) \quad (1)$$

where $B \subseteq A \subseteq \Omega$. Next, functions with monotonically decreasing marginal gains are *submodular*, satisfying the following inequality

$$g(A \cup C) - g(A) \leq g(B \cup C) - g(B) \quad (2)$$

³Although [16] introduces the method, [18] introduces the term RSP.

⁴Corah and Michael [16, Fig. 1] also describe a different mutual information objective which violates requirements for “coverage-like” objectives.

⁵Source code for the target tracking simulations is available at: <https://github.com/mcorah/MultiAgentSensing>

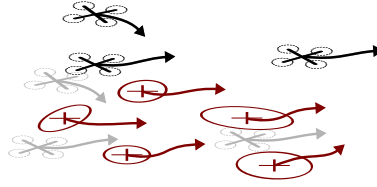


Fig. 1: A team of aerial robots \mathcal{R} (black) plan over a receding horizon to track a number of targets \mathcal{T} (red). The robots select sensing actions to minimize uncertainty in the target states which evolve independently of each other and the robots.

where $C \in \Omega \setminus A$. The differences in (2) express discrete derivatives of g . Drawing on notation for mutual information [28], the n^{th} derivative of g at $X \subseteq \Omega$ with respect to disjoint sets $Y_1, \dots, Y_n \subseteq \Omega$ can be defined recursively as

$$g(Y_1; \dots; Y_n | X) = g(Y_1; \dots; Y_{n-1} | X \cup Y_n) - g(Y_1; \dots; Y_{n-1} | X \cap Y_n) \quad (3)$$

where $g(X) = g(\cdot | X)$ is the 0^{th} derivative. So, (2) can be written as $g(C | A) \leq g(C | B)$, and so both monotonicity (1) and submodularity (2) form monotonicity conditions on derivatives of g [18, 29]. Further, second derivatives are written as⁶ $g(A; B | C) = g(A | B \cup C) - g(A | C)$ which expresses *effective redundancy* between A and B .

This text also abuses notation for sets and set functions: writing arguments to set functions like multivariate functions so $g(A, B) = g(A \cup B)$; implicitly wrapping elements $x \in \Omega$ in sets $g(x) = g(\{x\})$; writing integer ranges with the notation $i:j = \{k \mid i \leq k \leq j, k \in \mathbb{Z}\}$; and indexing into sets with sets of integers in subscripts as in $A_{1:5}$ where intent is clear.

B. Partition matroids for multi-robot systems

Each robot $i \in \mathcal{R}$ in a team $\mathcal{R} = \{1, \dots, n_r\}$ has access to a unique set of local control actions \mathcal{U}_i (i.e. the set of feasible receding-horizon trajectories). These sets of control actions are disjoint and together form (and partition) the set of all available actions $\Omega = \bigcup_{i \in \mathcal{R}} \mathcal{U}_i$. Each robot may choose any one action from its local set, and the set of all such complete and incomplete assignments forms a simple partition matroid $\mathcal{J} = \{X \subseteq \Omega \mid 1 \geq |X \cap \mathcal{U}_i|, \forall i \in \mathcal{R}\}$ [12, Sec. 39.4].

III. TARGET TRACKING PROBLEM

Consider a set of moving targets $\mathcal{T} = \{1, \dots, n_t\}$ and robots tracking those targets $\mathcal{R} = \{1, \dots, n_r\}$, seeking to minimize uncertainty (i.e. entropy [28]), as illustrated in Fig. 1. Let $\mathbf{x}_{i,t}^r \in \mathbb{R}^{d^r}$ and $\mathbf{x}_{j,t}^t \in \mathbb{R}^{d^t}$ be the respective states of robot $i \in \mathcal{R}$ and target $j \in \mathcal{T}$ at time $t \in \{0, \dots, T\}$. The states of each evolve in discrete time, with known dynamics

$$\mathbf{x}_{i,t+1}^r = f^r(\mathbf{x}_{i,t}^r, u_{i,t}), \quad \mathbf{x}_{j,t+1}^t = f^t(\mathbf{x}_{j,t}^t, \epsilon_{j,t}^t), \quad (4)$$

where $u_{i,t} \in \mathcal{U}$ is a control input from a finite set of inputs \mathcal{U} and $\epsilon_{j,t}^t$ is the targets' process noise. The robots then receive noisy observations $\mathbf{y}_{i,j,t}$ of the target states via

$$\mathbf{y}_{i,j,t} = h(\mathbf{x}_{i,t}^r, \mathbf{x}_{j,t}^t, \epsilon_{i,j,t}^y) \quad (5)$$

where $\epsilon_{i,j,t}^y$ is the observation noise. We refer to states and observations collectively with boldface capitals as \mathbf{X}_t^r , \mathbf{X}_t^t , and \mathbf{Y}_t , each at time t .

⁶We ignore intersections in (3) as variables are disjoint.

A. Receding-horizon optimization problem

Every so often, the robots plan to jointly maximize information gain over a receding horizon, starting at time t and with duration l , in what we refer to as a planning *epoch*. Specifically, robots maximize a submodular, monotonic, normalized objective g subject to a partition matroid constraint so that the optimal set of control actions is

$$X^* \in \arg \max_{X \in \mathcal{J}} g(X). \quad (6)$$

The partition matroid \mathcal{J} (Sec. II-B) represents assignment of sequences of control actions to robots with local sets

$$\mathcal{U}_i = \{(i, u_{1:l}) \mid u_{1:l} \in \mathcal{U}^l\} \quad \forall i \in \mathcal{R}. \quad (7)$$

The objective g is the mutual information between observations and target states given the choice of control actions. Now, interpret future states $\mathbf{X}_{t+1:t+l}^t$ and observations $\mathbf{Y}_{t+1:t+l}$ (taking care not to confuse states \mathbf{X} with sets X) as random variables induced by the process (4) and observation (5) noise terms. Then, writing future observations⁷ as $\mathbf{Y}_{t+1:t+l}(X)$ for $X \subseteq \Omega$, the mutual information is

$$g(X) = \mathbb{I}(\mathbf{X}_{t+1:t+l}^t; \mathbf{Y}_{t+1:t+l}(X) \mid \mathbf{Y}_{0:t}, \mathbf{X}_{0:t}^r) \quad (8)$$

where $\mathbb{I}(X; Y \mid Z)$ is the Shannon mutual information between X and Y conditional on Z and quantifies the reduction in uncertainty (entropy) of one random variable from observing another. We refer interested readers to Cover and Thomas [28] for more detail and thorough definitions. Crucially, mutual information objectives are normalized, monotonic, and submodular (defined in Sec. II-A) when observations are conditionally independent of target states [11] as they are here. However, mutual information does not satisfy the higher-order monotonicity condition which our prior work employs [16]. Instead, this work takes advantage of properties of a factored form of the objective. Equation (8) can be factored as a sum over the targets as

$$g(X) = \sum_{j \in \mathcal{T}} \mathbb{I}(\mathbf{X}_{j,t+1:t+l}^t; \mathbf{Y}_{j,t+1:t+l}(X) \mid \mathbf{Y}_{j,0:t}, \mathbf{X}_{j,0:t}^r) \quad (9)$$

because the target states and observations of the same are jointly independent and because the robot dynamics are deterministic. This follows because the mutual information is a difference of entropies [28, Eq. 2.45] which each decompose as sums over targets [28, Theorem 2.6.6].

B. Channel capacities and spatial locality

Spatial locality in target tracking problems arises from the factored form of the objective and how robots' capacities to sense targets decrease with distance. Variations in robots' $i \in \mathcal{R}$ abilities to sense different targets $j \in \mathcal{T}$ produce this spatial locality which take the form of channel capacities

⁷ $\mathbf{Y}_{t+1:t+l}(X)$ is a random variable expressing the noisy observations (5) that robot i receives after executing $u_{1:l}$ for each $(i, u_{1:l}) \in X$. For the purpose of analysis, $X \subseteq \Omega$ may include multiple assignments to one robot which will be associated with unique observation noise terms e^Y .

$C_{i,j}$ from information theory [28, Chapter 7] so that

$$C_{i,j} = \max_{x \in \mathcal{U}_i} \mathbb{I}(\mathbf{X}_{j,t+1:t+l}^t; \mathbf{Y}_{j,t+1:t+l}(x) \mid \mathbf{Y}_{j,0:t}, \mathbf{X}_{j,0:t}^r). \quad (10)$$

This channel capacity is an *upper bound on the amount of information a robot may obtain about a given target*. These channel capacities themselves form informative planning problems which we solve for the simulation results.

C. Computational model

A common feature of distributed planning problems is limited access to information. We assume for our computational model that each robot $i \in \mathcal{R}$ is able to approximate the objective for its own set of actions \mathcal{U}_i . That is, each robot has access to an approximation of marginal gains $\tilde{g}_i(x_i \mid A)$ which is valid for only its own actions $x_i \in \mathcal{U}_i$ and given any prior selections $A \subseteq \Omega$. This expresses both how robots evaluate mutual information approximately and how limited access to sensor data for distant targets could prohibit accurate evaluation of marginal gains for distant robots.

Robots likewise have limited access to the set of all control actions Ω : robots do not have access to others' actions except those they obtain by communicating each others' decisions, and robots also only obtain elements of their local sets $\mathcal{U}_i \subseteq \Omega$ implicitly via local planners.

IV. APPROACH TO DISTRIBUTED PLANNING

The proposed distributed planning framework seeks to approximate sequential solvers that are known to be near-optimal [19] but become impractical at large scales. This motivates our distributed approach which is similar but utilizes modifications and approximations to achieve planning in constant time and to satisfy constraints on information access for online, receding horizon planning. Later, Sec. V will describe the costs of these approximations, and Sec. VI will integrate these costs into a suboptimality guarantee that relates our distributed planner to sequential planning.

A. Greedy, sequential planning (an idealized planner)

The local greedy algorithm, originated by Fisher et al. [19] and applied to robotics by Singh et al. [13] enables robots to obtain near-optimal solutions to sensing problems by planning in sequence conditional on prior decisions. Robots produce solutions $X^g = \{x_1^g, \dots, x_{n_r}^g\}$ by greedy planning

$$x_i^g \in \arg \max_{x \in \mathcal{U}_i} g(x \mid X_{1:i-1}^g). \quad (11)$$

This local greedy algorithm is specific to partition matroids (Sec. II-B) and differs subtly from greedy algorithms for general matroids [19, 30, 31] which maximize over all possible actions (Ω) at each greedy step rather than robots' local control actions (\mathcal{U}) as in (11). Still, both greedy algorithms satisfy a constant-factor bound $g(X^g) \geq \frac{1}{2}g(X^*)$. While the local algorithm completes only one complete pass over Ω rather than one per robot, both require robots to make decisions in sequence: *the duration of planning for (11) is at least proportional to the size of the team*.

B. Distributed planning algorithm

Algorithm 1 provides pseudo-code for a distributed planner which can run in a constant number of sequential steps and produces distributed solutions X^d . As Fig. 2 illustrates, this planner follows a directed acyclic graph structure where robots are nodes and an edge to a robot represents access to another's decision. Designing the graph appropriately—ignoring some decisions—enables robots to plan in parallel.

Starting in lines 4–6, each robot begins planning upon receiving decisions from its in-neighbors $\mathcal{N}_i^{\text{in}}$. Robots approximate the objective given available computational resources and sensor data θ_i via \tilde{g}_i . The available sensor data may include data for all targets or only those near the robot (we will study both cases). We assume robots obtain this sensor data via inter-robot communication (not shown). Then, in lines 7–8, once the planner exits or runs out of time, the robot commits to an action which it executes in a receding-horizon fashion and sends to any out-neighbors $\mathcal{N}_i^{\text{out}}$.

As described, planning proceeds asynchronously, and robots have access to both $\mathcal{N}_i^{\text{in}}$ and $\mathcal{N}_i^{\text{out}}$. However, [18, Chapter 8] provides a more practical time-synchronous implementation where the graph structure is implicit.

1) *Planning in parallel via RSP*: To enable parallel planning with little impact on suboptimality, robots construct the directed graph structure via Randomized Sequential Partitions (RSP) [16]. When planning via RSP robots assign themselves randomly to one of n_d sequential steps. Then, robots assigned to the same step plan in parallel with access (via $\mathcal{N}_i^{\text{in}}$) to some or all decisions from prior steps.

The cost of planning via RSP instead of sequentially (11) approaches zero when effective redundancy between agents is bounded [16]. In Sec. VI, we will obtain a new such bound for target tracking in terms of a weighted undirected graph $\mathcal{G} = (\mathcal{R}, \mathcal{E}, \widehat{\mathcal{W}})$ which connects the robots with edges $\mathcal{E} = \{(i, j) | i, j \in \mathcal{R}, i \neq j\}$ whereas [16] proves that when the optimum is proportional to the sum of weights:

$$g(X^*) \propto \sum_{(i,j) \in \mathcal{E}} \widehat{\mathcal{W}}(i, j) \quad (12)$$

distributed planning with a constant number of steps guarantees suboptimality approaching half of optimal (with $1/n_d$)

Algorithm 1 Distributed algorithm for receding-horizon target tracking from the perspective of robot $i \in \mathcal{R}$ for execution at time t .

- 1: $\mathcal{N}_i^{\text{in}} \leftarrow$ in-neighbors of robot i
 - 2: $\mathcal{N}_i^{\text{out}} \leftarrow$ out-neighbors of robot i
 - 3: $\theta_i \leftarrow$ sensor data (or summary) accessible to robot i
 - 4: RECEIVE: $X_{\mathcal{N}_i^{\text{in}}}^d$ from $\mathcal{N}_i^{\text{in}}$
 - 5: $\tilde{g}_i \leftarrow$ approximation of g given θ_i
 - 6: $x_i^d \leftarrow \text{PLANANYTIME}(\tilde{g}_i, \mathbf{x}_{i,t}^r, X_{\mathcal{N}_i^{\text{in}}}^d)$
 - 7: SEND: x_i^d to $\mathcal{N}_i^{\text{out}}$
 - 8: EXECUTE: x_i^d starting at time t and until the beginning of the result of the next planning round
-

in expectation, for any number of robots [16, Theorem 3].

V. COST MODEL FOR APPROXIMATE PLANNING

This section describes the costs of planning with directed acyclic graphs via RSP and of approximations in planning and objective evaluation given constraints on computation time and information access. Rather than assume constant-factor suboptimality at each step [13]—as would be appropriate if the local planner satisfied a consistent performance guarantee—we present this flexible cost model to account for uncertainty arising from planning in real time. The analysis (Sec. VI) will integrate these costs for *distributed planning*, *objective evaluation*, and *anytime (single-robot) planning* into a suboptimality guarantee that relates the suboptimality of Alg. 1 to the bound for sequential planning (11).

A. General cost of suboptimal decisions for individual robots

Before describing the specific costs of approximations, let us define a general cost in terms of the difference between the exact marginal gain for a greedy decision and the gain for the actual decision produced by the planner x_i^d . Specifically, given an instance of (6) with objective g and a subset of prior decisions $X \subseteq X_{1:i-1}^d$, the cost to robot $i \in \mathcal{R}$ for making a suboptimal decision x_i^d is the difference between the utility of that decision and the true maximum over \mathcal{U}_i

$$\gamma_i^{\text{gen}}(g, x_i^d, X) = \max_{x \in \mathcal{U}_i} g(x|X) - g(x_i^d|X). \quad (13)$$

This expression will be useful both for defining specific costs and as a tool for analyzing suboptimality.

B. Cost of distributed planning on directed acyclic graphs

Planning nominally via the greedy algorithm (11), robot $i \in \mathcal{R}$ has access to prior decisions by robots $\{1, \dots, i-1\}$. In our approach, (Alg. 1) robots only have access to a subset $\mathcal{N}_i^{\text{in}} \subseteq \{1, \dots, i-1\}$ of these decisions, which induces a directed acyclic graph with edges (j, i) for each robot $j \in \mathcal{N}_i^{\text{in}}$ whose decision i accesses while planning [25, 26]. In a sense, the robots ignore decisions by $\widehat{\mathcal{N}}_i^{\text{in}} = \{1, \dots, i-1\} \setminus \mathcal{N}_i^{\text{in}}$, and the cost of doing so is a second derivative (3):

$$\gamma_i^{\text{dist}} = g(x_i^d | X_{\mathcal{N}_i^{\text{in}}}^d) - g(x_i^d | X_{1:i-1}^d) = -g(x_i^d, X_{\widehat{\mathcal{N}}_i^{\text{in}}}^d | X_{\mathcal{N}_i^{\text{in}}}^d). \quad (14)$$

This cost expresses the effective redundancy between i 's decision and the decisions by $\widehat{\mathcal{N}}_i^{\text{in}}$ which were ignored. Later, we will upper bound this redundancy in terms of $\widehat{\mathcal{N}}_i^{\text{in}}$ and eliminate the dependency on $X_{\mathcal{N}_i^{\text{in}}}^d$.

C. Cost of approximate evaluation of the objective

We assume robots can access relevant data to evaluate the mutual information objective (8), either exactly or approximately such as by ignoring distant targets or via sampling. Either way, robot $i \in \mathcal{R}$ has access to a local approximation \tilde{g}_i of the objective, and the cost of this approximation (treating stochasticity implicitly) is at most the sum of the

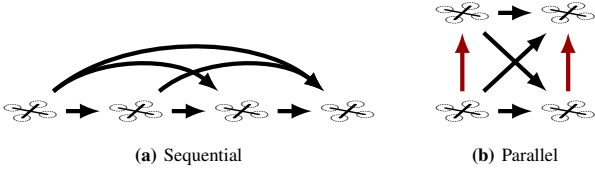


Fig. 2: Graph models for two distributed planners where edges represent access to robots' prior decisions. (a) The sequential greedy algorithm (11) corresponds to a complete directed acyclic graph because robots have access to all prior decisions. This graph has one sequential step per robot (time increases toward the right). (b) Deleting edges (red) removes temporal constraints due to requiring access to prior decisions. Here, deleting edges enables pairs of robots to plan in parallel, and the planner subsequently requires two sequential steps instead of four.

maximum over- and under-approximation of g

$$\gamma_i^{\text{obj}} = \max_{x_1, x_2 \in \mathcal{U}_i} \left(\tilde{g}_i(x_1 | X_{\mathcal{N}_i^{\text{in}}}^{\text{d}}) - g(x_1 | X_{\mathcal{N}_i^{\text{in}}}^{\text{d}}) \right. \\ \left. + g(x_2 | X_{\mathcal{N}_i^{\text{in}}}^{\text{d}}) - \tilde{g}_i(x_2 | X_{\mathcal{N}_i^{\text{in}}}^{\text{d}}) \right) \quad (15)$$

where x_1 and x_2 are the points where \tilde{g}_i most under- and over-approximates g over the local action set \mathcal{U}_i .

D. Cost of approximate (anytime) single-robot planning

Selecting sensing actions to maximize information gain for an individual robot over a finite horizon produces an informative path planning problem [13, 22]. However, robots have limited amounts of time available and must terminate planning and transmit results soon enough so that later robots that depend on those decisions can make their own decisions in time for the plans go into effect (at time t in Algorithm 1).

Although some existing planners provide performance guarantees [13, 22, 32], designers applying these methods may have to vary replanning rates or tune problem parameters to satisfy constraints on planning time for operation in real-time. On the other hand, randomized planners [23, 24] and gradient- and Newton-based trajectory generation [33, 34] converge to local or global maxima but typically provide no guarantees on solution quality before convergence for anytime planning. Likewise, this paper applies Monte Carlo tree search [35, 36], a common randomized planner for single-robot planning (see the results, Sec. VIII).

We model online planners via their empirical performance at approximating optimal single-robot solutions conditional on $X_{\mathcal{N}_i^{\text{in}}}^{\text{d}}$ and with the local objective \tilde{g}_i

$$\gamma_i^{\text{plan}} = \gamma_i^{\text{gen}}(\tilde{g}_i, x_i^{\text{d}}, X_{\mathcal{N}_i^{\text{in}}}^{\text{d}}). \quad (16)$$

This captures the inherent uncertainty in anytime planning and enables us to characterize collective performance in terms of the bulk suboptimality of single-robot planning.

VI. ANALYSIS OF SUBOPTIMALITY OF DISTRIBUTED PLANNING

This section analyzes suboptimality for distributed planning. Specifically, Alg. 1 achieves a performance bound which approaches that for sequential planning (Sec IV-A) with suboptimality arising from the aforementioned costs.

Theorem 1 (Suboptimality of Alg. 1): Considering an instance of (6), any solution X^{d} that Alg. 1 produces satisfies

$$g(X^*) \leq 2g(X^{\text{d}}) + \sum_{i \in \mathcal{R}} \left(\gamma_i^{\text{dist}} + \gamma_i^{\text{obj}} + \gamma_i^{\text{plan}} \right), \quad (17)$$

and the total cost of distributed planning is bounded by

$$\sum_{i \in \mathcal{R}} \gamma_i^{\text{dist}} \leq \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^{\text{in}}} \widehat{\mathcal{W}}(i, j) \quad (18)$$

where $\widehat{\mathcal{W}}(i, j)$ is an edge weight that bounds effective redundancy between pairs of robots for observing the same targets (whose expression we provide in following analysis). *This sum of weights approaches zero* when planning via RSP with increasing numbers of rounds n_{d} (Sec. IV-B.1).

The proof of Theorem 1 is in Appendix III. We summarize this result in (Sec. VI-C) after introducing preliminary results related to the first (Sec. VI-A) and second (Sec. VI-B) parts of the theorem.

Regarding the form of Theorem 1, this bound characterizes practical implementations of Alg. 1. An idealized version of our distributed algorithm would obtain exact objective values and maxima, and the associated costs γ^{obj} and γ^{plan} would each be zero. From this perspective, (17) describes how real implementations may deviate from this ideal and states that suboptimality arises as an accumulation of individual inefficiencies which can be modeled empirically.

A. General suboptimality in multi-robot planning

The following lemma expresses the joint suboptimality of any solution as a sum of costs of suboptimal decisions.

Lemma 2 (Suboptimality of general assignments): Given some submodular, monotonic, normalized objective g , any assignment of actions to all robots (a basis) $X^{\text{d}} \in \mathcal{J}$ on a simple partition matroid satisfies

$$g(X^*) \leq 2g(X^{\text{d}}) + \sum_{i=1}^{n_r} \gamma_i^{\text{gen}}(g, x_i^{\text{d}}, X_{1:i-1}^{\text{d}}). \quad (19)$$

The proof of Lemma 2 is in Appendix II. Observe that if we obtain X^{d} via exact sequential maximization, the summands (γ^{gen}) of (19) are zero, and we obtain the original result by Fisher et al. [19] and likewise for single-robot solvers with constant-factor suboptimality [13, Theorem 1].

B. Bounding the cost of distributed planning for target tracking problems

This section characterizes the cost of distributed planning (18) in target tracking problems. We begin by investigating the decomposition of the objective as a sum over targets to obtain an intermediate bound. Applying this bound produces the weights $\widehat{\mathcal{W}}$ which relate the cost of ignoring robots during distributed planning (Sec. V-B) to the channel capacities (10) between the robots and targets.

1) *Decomposing objectives as sums:* The objective (8) for the target tracking problem we study forms a sum over information sources, the targets (9). Let $\mathcal{G} = \{g_1, \dots, g_{n_t}\}$ be a collection of set functions so that for $j \in \mathcal{T}$ and $X \subseteq \Omega$

$$g_j(X) = \mathbb{I}(X_{j,t+1:t+l}^{\text{t}}; \mathbf{Y}_{j,t+1:t+l}(X) | \mathbf{Y}_{j,0:t}, \mathbf{X}_{0:t}^{\text{r}}). \quad (20)$$

This collection of set functions \mathcal{G} decomposes g and enables us to characterize interactions between robots in terms of robots' capacities to sense near and distant targets.

Definition 1 (Sum decomposition): A set of submodular, monotonic, normalized functions $\mathcal{G} = \{g_1, \dots, g_n\}$ decomposes a set function g if

$$g(X) = \sum_{\hat{g} \in \mathcal{G}} \hat{g}(X), \quad \text{for all } X \subseteq \Omega. \quad (21)$$

Closure over sums [29] ensures that g is submodular, monotonic, and normalized if the same is true for each $\hat{g} \in \mathcal{G}$. Further, although some such sum decomposition always exists ($\mathcal{G} = \{g\}$), the choice of decomposition affects the tightness of the performance bound; choosing \mathcal{G} to form a sum over targets (20) will capture spatial locality in distributions of robots and targets.

2) *Derivatives and the sum decomposition:* Given some decomposition \mathcal{G} of g , the second derivative (3) of g at $X \subseteq \Omega$ with respect to $A, B \subseteq \Omega$, all disjoint, is

$$g(A; B|X) = \sum_{\hat{g} \in \mathcal{G}} \hat{g}(A; B|X). \quad (22)$$

This derivative has the form of the negation of the cost of ignoring decisions during distributed planning γ^{dist} (14), and the rest of this section is devoted to obtaining a bound on expressions which relate a robot's decision A to the decisions that robot ignores B while eliminating dependency on which prior decisions the robot has access to X .

3) *Bounding second derivatives via sum decompositions:* Applying monotonicity and submodularity respectively provides a lower bound on the second derivative of a set function

$$g(A; B|X) = g(A|B, X) - g(A|X) \geq -g(A|X) \geq -g(A) \quad (23)$$

where $A, B, X \subseteq \Omega$ are disjoint. By symmetry

$$g(A; B|X) \geq -\min(g(A), g(B)). \quad (24)$$

Then, expressing the second derivative of g in terms of the sum decomposition (22) and using (24) to bound the derivatives of $\hat{g} \in \mathcal{G}$ yields

$$g(A; B|X) \geq \sum_{\hat{g} \in \mathcal{G}} -\min(\hat{g}(A), \hat{g}(B)). \quad (25)$$

Remark 1: Our prior work [16] relies on $g(A; B|X)$ increasing monotonically in X to state $g(A; B|X) \geq g(A; B)$. To unify this result with (25), we state that each produces a lower bound on $g(A; B|X)$ as a function of A and B .

4) *Quantifying inter-robot redundancy:* This bound on the second derivative of g (25) leads to a bound on redundancy between agents which we express with the weights:

$$\begin{aligned} \mathcal{W}(i, j) &= \max_{x_i \in \mathcal{U}_i, x_j \in \mathcal{U}_j} \sum_{\hat{g} \in \mathcal{G}} \min(\hat{g}(x_i), \hat{g}(x_j)) \\ &\geq -g(x'_i; x'_j|X) \\ &\quad \text{for all } x'_i \in \mathcal{U}_i, x'_j \in \mathcal{U}_j, X \subseteq \Omega \setminus \{x'_i, x'_j\}. \end{aligned} \quad (26)$$

Evaluating values of \mathcal{W} is difficult as doing so involves search over the product of two robots' action spaces. To make

evaluation of weights tractable, relaxing this expression by taking the pairwise minimum of the maximum values of each objective component produces an upper bound in terms of channel capacities (10), avoiding search over a product space

$$\begin{aligned} \widehat{\mathcal{W}}(i, j) &= \sum_{k \in \mathcal{T}} \min(C_{i,k}, C_{j,k}) \\ &= \sum_{\hat{g} \in \mathcal{G}} \min\left(\max_{x_i \in \mathcal{U}_i} \hat{g}(x_i), \max_{x_j \in \mathcal{U}_j} \hat{g}(x_j)\right) \\ &\geq \max_{x_i \in \mathcal{U}_i, x_j \in \mathcal{U}_j} \sum_{\hat{g} \in \mathcal{G}} \min(\hat{g}(x_i), \hat{g}(x_j)) = \mathcal{W}(i, j). \end{aligned} \quad (27)$$

The second equality follows from the definition of the channel capacities (10), recalling that we chose \mathcal{G} to decompose g by targets \mathcal{T} . This expresses how *if sensing capacity decreases with distance so do interactions between robots.*

C. Summary of the proof of Theorem 1

Theorem 1 consists of two parts. The first, the effect of approximations on planning performance (17) follows by applying Lemma 2, on the suboptimality of general assignments, and substituting the definitions of the costs ((14), (15), and (16)). The second part (18) characterizes suboptimality due to distributed planning and follows by applying a chain rule (Appendix I) to the definition of cost of distributed planning (14) and substituting (25), (26), and (27). Please refer to Appendix III for the full proof.

VII. RUN TIME AND SCALING

Algorithm 1 requires a number of sequential planning steps that depends on the planner graph (Sec. V-B), a constant number of steps for RSP. Further, Sec. IV-B.1 stated that if the optimum is proportional to the sum of weights (12) RSP guarantees suboptimality approaching half of optimal.

Assuming the optimum is proportional to the number of robots (n_r), then (12) holds if the sum of weights is proportional to n_r as well. Appendix IV presents sufficient conditions for the sum of weights to be proportional to n_r , given the robot-target channel capacities (10) are bounded appropriately with distance. This ensures the cost of distributed planning γ^{dist} (14) is bounded independent of n_r .

Additionally, bounds on channel capacities as a function of distance enable robots to ignore sufficiently distant robots and targets in what we refer to as Range-limited RSP (R-IRSP) [18] as the additional cost (ignoring decisions and approximating the objective) approaches zero.

By incorporating range limits and tracking targets with sparse Bayes filters we also achieve single-robot planning in constant time (depending on densities of robots and targets).

Regarding communication, robots send one message with constant size for each edge in the directed planner graph (Sec. V-B). Ignoring distant robots via R-IRSP reduces the total to a constant number of messages per robot [16].

VIII. RESULTS

To evaluate the approach, we provide simulation results (visualized in Fig. 3) for teams of robots tracking groups

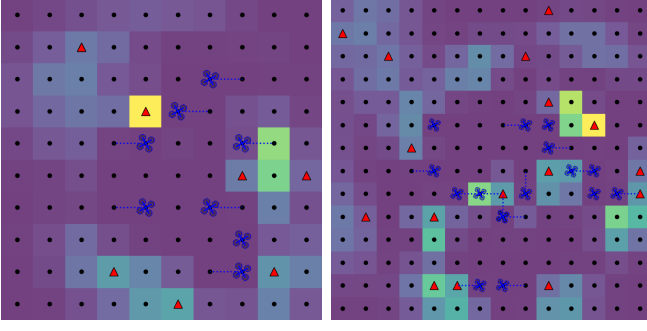


Fig. 3: Visualizations of eight and sixteen robots tracking same numbers of targets. Robots with dotted finite-horizon trajectories are blue and targets red. The background illustrates the sum of target probabilities at each grid space, increasing from purple to yellow.

of targets (one target per robot). Robots move according to planner output and targets via a random walk, all on a square four-connected grid with $\sqrt{12.5n_r}$ cells on each side.⁸ The robots estimate target locations via Bayesian filters⁹ given range observations to each target with mean $\hat{d} = \min(d, 20)$ and variance $0.25 + 0.5\hat{d}^2$ where d is the Euclidean distance to the target in cells lengths.¹⁰ For the purpose of this paper, robots have access to all observations or, equivalently, centralized filters. Trials run for 100 time-steps; initial states are uniformly random; initial target positions are known,¹¹ and we ignore the first 20 steps of each trial to allow the system to converge to steady-state conditions.

Robots plan actions individually using Monte Carlo tree search (MCTS, PLANANYTIME) [36] with a two step horizon and collectively according to the specified distributed planner. To ensure tractability we replace the original objective (8) with a sum of mutual information for each time-step

$$g^{\text{sim}}(X) = \sum_{k=1}^l \mathbb{I}(\mathbf{x}_{t+k}^t; \mathbf{Y}_{t+1:t+k}(X) | \mathbf{Y}_{0:t}, \mathbf{x}_{0:t}^r), \quad X \subseteq \Omega. \quad (28)$$

This objective is equivalent to [37, (18)] and can be thought of as minimizing uncertainty at the time of each planning step. Being a sum, (28) remains submodular, monotonic, and normalized, and all analysis, including Theorem 1, applies unchanged. Like Ryan and Hedrick [37], we evaluate this objective by simulating the system and computing the sample mean of filter entropy. The MCTS planner also estimates the objective implicitly by simulating the system once per rollout; by sampling more valuable actions more often, MCTS produces increasingly accurate estimates for nearly optimal trajectories. The experiments compare methods for multi-robot coordination including: sequential planning (11);

⁸Numbers of targets and grid cells are proportional to the number of robots (n_r). We desire entropies approaching a constant for large n_r on a per-robot basis and the same for redundancy and objective values.

⁹Additionally, planners with sixteen or more robots use sparse filters, ignoring target occupancy probabilities below 10^{-3} .

¹⁰We selected simulation parameters to maximize discrepancy between sequential and myopic planning without evaluating the performance of RSP.

¹¹Known initial states promote fast convergence and ensure initial uncertainty does not increase with environment size.

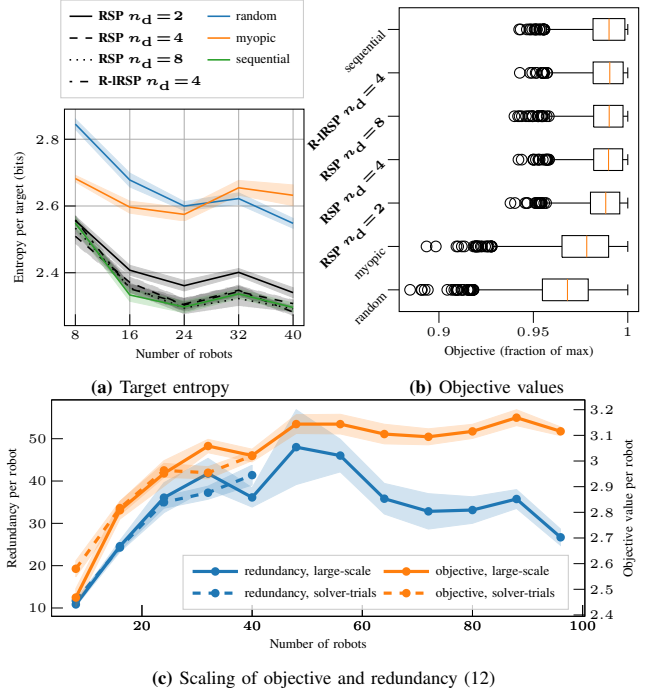


Fig. 4: (a) Regarding target entropy, the task performance criterion, (lower is better) our RSP planners consistently improve upon myopic planning and approach sequential planning with many times fewer sequential steps. (b) Objective values on common 16-robot subproblems reflect a similar trend, and results for distributed planning (bold) closely match sequential. (c) Considering average objective values and total redundancy (12) (both per robot) for RSP with $n_d = 4$ (including values for trials from (b) and five additional trials with R-IRSP with up to 96 robots), values initially increase but appear to approach asymptotes, indicating that the two are approximately proportional at large scales (12) so that RSP planning approaches constant-factor suboptimality. Shaded regions depict standard error.

distributed planning (Alg. 1) with RSP [16]; myopic planning (MCTS without coordination); and random actions. Additionally, we provide results with range limits (R-IRSP) where robots ignore targets further than 12 units (in terms of the filter mean positions) and robots further than 20 units. Given use of sparse filters, R-IRSP runs in *constant time*.

We evaluate the distributed planning approach via *task performance* (average target entropy) for various numbers of robots (Fig. 4), *objective values* on a common set of subproblems (6), and the *redundancy per robot* (the sum of weights (12) divided by n_r). The results for average target entropy—which express uncertainty in target locations [28]—are based on 20 simulations of target tracking for each configuration. Results for objective values and redundancy use planning subproblems (6) taken from the simulation trials for RSP with $n_d = 4$. The results for objective values by solver are for 16 robots and are normalized according to maximum values across solvers for each planning problem. For results on redundancy, an additional five trials of R-IRSP with $n_d = 4$ demonstrate behavior for up to 96 robots.¹²

Proposed distributed planners provide consistent improvements in target tracking performance (average target entropy) (Fig. 4a) compared to myopic planning; distributed planning

¹²Planning at this scale is intractable for other planner configurations.

in eight rounds matches sequential planning despite requiring as much as five times fewer planning steps and produces 5–13% better (lower) target entropy than when planning myopically. The objective values (Fig. 4b) exhibit a similar trend, and all distributed planners closely match sequential planning. Although the objective values and redundancy per robot (Fig. 4c) initially increase with more robots, the values eventually level off and are roughly proportional (12). This is consistent with analysis of scaling behavior in Appendix II which suggests that redundancy per robot approaches a constant value. Overall, the results indicate that even a small amount of coordination ($n_d = 2$), independent of the number of robots, is sufficient to provide performance comparable to sequential planning in receding-horizon settings.

IX. CONCLUSIONS AND FUTURE WORK

This paper has presented a distributed planner for mutual information-based target tracking which runs in a fixed number of steps and mitigates growth in planning time for existing sequential planners for submodular maximization. The analysis provided a novel bound on suboptimality by using target independence to decompose the objective as a sum. Additionally, by explicitly accounting for suboptimal local planning (e.g. anytime planning) and approximation of the objective, we affirmed that the proposed approach is applicable to practical tracking systems. The results demonstrated that distributed planning improves tracking performance (in terms of target entropy) compared to planners with no coordination and that distributed planning with little coordination can even match fully sequential planning given a constant number of planning rounds. Finally, although we focused on target tracking, future work may take advantage of how the analysis applies to general multi-objective sensing problems, possibly in concert with our work on coverage [16].

REFERENCES

- [1] O. M. Cliff, R. Fitch, S. Sukkarieh, D. L. Saunders, and R. Heinsohn, "Online localization of radio-tagged wildlife with an autonomous aerial robot system," in *Proc. of Robot.: Sci. and Syst.*, Rome, Italy, Jul. 2015.
- [2] K. Shah and M. Schwager, "GRAPE: Geometric risk-aware pursuit-evasion," *Robot. Auton. Syst.*, 2019.
- [3] B. Charrow, V. Kumar, and N. Michael, "Approximate representations for multi-robot control policies that maximize mutual information," *Auton. Robots*, vol. 37, no. 4, pp. 383–400, 2014.
- [4] B. Charrow, N. Michael, and V. Kumar, "Cooperative multi-robot estimation and control for radio source localization," *Intl. Journal of Robotics Research*, vol. 33, no. 4, pp. 569–580, Apr. 2014.
- [5] C. Piacentini, S. Bernardini, and J. C. Beck, "Autonomous target search with multiple coordinated UAVs," *J. Artif. Intell. Research*, vol. 65, pp. 519–568, 2019.
- [6] G. Hollinger, S. Singh, J. Djughash, and A. Kehagias, "Efficient multi-robot search for a moving target," *Intl. Journal of Robotics Research*, vol. 28, no. 2, pp. 201–219, 2009.
- [7] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proc. of the IEEE/CVF Conf. on Comput. Vis. and Pattern Recognition*, Long Beach, CA, Jun. 2019.
- [8] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Auton. Robots*, vol. 31, no. 4, p. 299, 2011.
- [9] P. Tokekar, V. Isler, and A. Franchi, "Multi-target visual tracking with aerial robots," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Chicago, IL, Sep. 2014.
- [10] L. Zhou and P. Tokekar, "Sensor assignment algorithms to improve observability while tracking targets," *IEEE Trans. Robotics*, vol. 35, no. 5, pp. 1206–1219, 2019.
- [11] 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, Jul. 2005.
- [12] A. Schrijver, *Combinatorial optimization: polyhedra and efficiency*. Springer Science & Business Media, 2003, vol. 24.
- [13] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *J. Artif. Intell. Res.*, vol. 34, pp. 707–755, 2009.
- [14] U. Feige, "A threshold of $\ln n$ for approximating set cover," *Journal of the ACM (JACM)*, vol. 45, no. 4, pp. 634–652, 1998.
- [15] 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.
- [16] M. Corah and N. Michael, "Distributed submodular maximization on partition matroids for planning on large sensor networks," in *Proc. of the IEEE Conf. on Decision and Control*, Miami, FL, Dec. 2018.
- [17] Y. Sung, A. K. Budhiraja, R. K. Williams, and P. Tokekar, "Distributed assignment with limited communication for multi-robot multi-target tracking," *Auton. Robots*, 2019.
- [18] M. Corah, "Sensor planning for large numbers of robots," Ph.D. dissertation, Carnegie Mellon University, 2020.
- [19] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions-II," *Polyhedral Combinatorics*, vol. 8, pp. 73–87, 1978.
- [20] S. Jorgensen, R. H. Chen, M. B. Milam, and M. Pavone, "The matroid team surviving orienteers problem: Constrained routing of heterogeneous teams with risky traversal," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Vancouver, Canada, Sep. 2017.
- [21] M. Corah and N. Michael, "Distributed matroid-constrained submodular maximization for multi-robot exploration: theory and practice," *Auton. Robots*, vol. 43, no. 2, pp. 485–501, 2019.
- [22] C. Chekuri and P. Martin, "A recursive greedy algorithm for walks in directed graphs," in *Proc. of the IEEE Annu. Symp. Found. Comput. Sci.*, 2005, pp. 245–253.
- [23] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *Intl. Journal of Robotics Research*, vol. 33, no. 9, pp. 1271–1287, 2014.
- [24] M. Lauri and R. Ritala, "Planning for robotic exploration based on forward simulation," *Robot. Auton. Syst.*, vol. 83, pp. 15–31, 2016.
- [25] B. Ghahesifard and S. L. Smith, "Distributed submodular maximization with limited information," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 4, pp. 1635–1645, 2017.
- [26] D. Grimsman, M. S. Ali, J. P. Hespanha, and J. R. Marden, "The impact of information in greedy submodular maximization," *IEEE Trans. Control Netw. Syst.*, 2018.
- [27] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, 2008.
- [28] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY: John Wiley & Sons, 2012.
- [29] S. Foldes and P. L. Hammer, "Submodularity, supermodularity, and higher-order monotonicities of pseudo-boolean functions," *Mathematics of Operations Research*, vol. 30, no. 2, pp. 453–461, 2005.
- [30] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Trans. Robotics*, vol. 25, no. 4, pp. 912–926, 2009.
- [31] R. K. Williams, A. Gasparri, and G. Ulivi, "Decentralized matroid optimization for topology constraints in multi-robot allocation problems," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Singapore, May 2017.
- [32] H. Zhang and Y. Vorobeychik, "Submodular optimization with routing constraints," in *Assoc. for Adv. of Artif. Intell.*, 2016.
- [33] 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, Jul. 2015.
- [34] V. Indelman, L. Carlone, and F. Dellaert, "Planning under uncertainty in the continuous domain: a generalized belief space approach," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Hong Kong, China, Jun. 2014.
- [35] 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*, vol. 4, no. 1, pp. 1–43, 2012.
- [36] G. Chaslot, "Monte-Carlo tree search," Ph.D. dissertation, Universiteit Maas-tricht, 2010.
- [37] A. Ryan and J. K. Hedrick, "Particle filter based information-theoretic active sensing," *Robot. Auton. Syst.*, vol. 58, no. 5, pp. 574–584, 2010.

APPENDIX I

THE CHAIN RULE FOR DERIVATIVES OF SET FUNCTIONS

Set functions and their derivatives satisfy chain rules analogous to those for entropy and mutual information [28, Theorem 2.5.1–2]. We provide a general statement here which we will apply to first and second derivatives of set functions.

Lemma 3 (Chain rule for derivatives of set functions):

Consider sets $Y_1, \dots, Y_n, X \subseteq \Omega$, all disjoint. Then, writing the elements of Y_n as $Y_n = \{y_{n,1}, \dots, y_{n,|Y_n|}\}$, the derivative of g can be rewritten in terms of derivatives with respect to the individual elements of Y_n as

$$g(Y_1; \dots; Y_n | X) = \sum_{i=1}^{|Y_n|} g(Y_1; \dots; Y_{n-1}; y_{n,i} | Y_{n,1:i-1}, X). \quad (29)$$

Proof: The proof follows by expanding the derivative (3), forming a telescoping sum, and rewriting the summands as individual derivatives:

$$\begin{aligned} g(Y_1; \dots; Y_n | X) &= g(Y_1; \dots; Y_{n-1} | Y_n, X) \\ &\quad - g(Y_1; \dots; Y_{n-1} | X) \\ &= \sum_{i=1}^{|Y_n|} (g(Y_1; \dots; Y_{n-1} | Y_{n,1:i}, X) \\ &\quad - g(Y_1; \dots; Y_{n-1} | Y_{n,1:i-1}, X)) \\ &= \sum_{i=1}^{|Y_n|} g(Y_1; \dots; Y_{n-1}; y_{n,i} | Y_{n,1:i-1}, X). \end{aligned} \quad (30)$$

APPENDIX II

PROOF OF LEMMA 2, SUBOPTIMALITY OF GENERAL ASSIGNMENTS

Proof: This result follows typical methods for sequential submodular maximization with slight changes to assist in book-keeping:

$$g(X^*) \leq g(X^d, X^*) \quad (31)$$

$$= g(X^d) + \sum_{i=1}^{n_r} g(x_i^* | X_{1:i-1}^d, X^d) \quad (32)$$

$$\leq g(X^d) + \sum_{i=1}^{n_r} g(x_i^* | X_{1:i-1}^d) \quad (33)$$

$$\leq g(X^d) + \sum_{i=1}^{n_r} \max_{x \in \mathcal{U}_i} g(x | X_{1:i-1}^d) \quad (34)$$

$$\leq g(X^d) + \sum_{i=1}^{n_r} (g(x_i^d | X_{1:i-1}^d) + \gamma_i^{\text{gen}}(g, x_i^d, X_{1:i-1}^d)) \quad (35)$$

$$= 2g(X^d) + \sum_{i=1}^{n_r} \gamma_i^{\text{gen}}(g, x_i^d, X_{1:i-1}^d). \quad (36)$$

Above, (31) follows from monotonicity; (32) expands a telescoping series; (33) follows from submodularity; (34)

upper bounds the gains for the optimal decisions x_i^* with the maximum marginal gains; (35) substitutes the expression for general suboptimality (13) thereby adding and subtracting the marginal gains for x_i^d ; and (36) collapses the telescoping series. ■

APPENDIX III

PROOF OF THEOREM 1, SUBOPTIMALITY OF DISTRIBUTED PLANNING

Proof: Theorem 1 consists of two parts, (17) and (18). We prove each in turn. Since the costs in both equations involve sums over robots, both proofs analyze costs with respect to some robot $i \in \mathcal{R}$.

1) *Proof of Theorem 1, part 1* (17): According to the standard greedy algorithm (11), robot i would plan conditional on decisions by robots $\{1, \dots, i-1\}$. However, in Alg. 1 that robot instead plans conditional on decisions by a subset of these robots $\mathcal{N}_i^{\text{in}} \subseteq \{1, \dots, i-1\}$ and ignores $\hat{\mathcal{N}}_i^{\text{in}} = \{1, \dots, i-1\} \setminus \mathcal{N}_i^{\text{in}}$. Recalling Lemma 2, we can write the suboptimality of decisions X^d in terms of a general cost γ_i^{gen} . Let us now extract the cost of distributed planning γ_i^{dist} from this expression as follows:

$$\begin{aligned} \gamma_i^{\text{gen}}(g, x_i^d, X_{1:i-1}^d) &\leq \max_{x \in \mathcal{U}_i} g(x | X_{\mathcal{N}_i^{\text{in}}}^d) - g(x_i^d | X_{1:i-1}^d) \\ &= \gamma_i^{\text{gen}}(g, x_i^d, X_{\mathcal{N}_i^{\text{in}}}^d) \\ &\quad + g(x_i^d | X_{\mathcal{N}_i^{\text{in}}}^d) - g(x_i^d | X_{1:i-1}^d) \\ &= \gamma_i^{\text{gen}}(g, x_i^d, X_{\mathcal{N}_i^{\text{in}}}^d) + \gamma_i^{\text{dist}}. \end{aligned} \quad (37)$$

Here, the first step follows by referring to the general cost model (13) and observing that $\max_{x \in \mathcal{U}_i} g(x | X_{1:i-1}^d) \leq \max_{x \in \mathcal{U}_i} g(x | X_{\mathcal{N}_i^{\text{in}}}^d)$ due to submodularity. The second rewrites the cost in terms of decisions with respect to $X_{\mathcal{N}_i^{\text{in}}}^d$; and the last substitutes the cost of distributed planning (14).

To incorporate the cost of suboptimal planning γ_i^{plan} , observe that

$$\begin{aligned} \gamma_i^{\text{gen}}(g, x_i^d, X_{\mathcal{N}_i^{\text{in}}}^d) &= \gamma_i^{\text{gen}}(g, x_i^d, X_{\mathcal{N}_i^{\text{in}}}^d) + \gamma_i^{\text{plan}} - \gamma_i^{\text{plan}} \\ &= \gamma_i^{\text{plan}} + \gamma_i^{\text{gen}}(g, x_i^d, X_{\mathcal{N}_i^{\text{in}}}^d) \\ &\quad - \gamma_i^{\text{gen}}(\tilde{g}_i, x_i^d, X_{\mathcal{N}_i^{\text{in}}}^d) \end{aligned} \quad (38)$$

which follows from the definition of the planning cost (16).

The cost of approximation of the objective γ_i^{obj} upper bounds the difference of the last two terms in (38):

$$\begin{aligned} \gamma_i^{\text{gen}}(g, x_i^d, X_{\mathcal{N}_i^{\text{in}}}^d) - \gamma_i^{\text{gen}}(\tilde{g}_i, x_i^d, X_{\mathcal{N}_i^{\text{in}}}^d) &= \tilde{g}_i(x_i^d | X_{\mathcal{N}_i^{\text{in}}}^d) - g(x_i^d | X_{\mathcal{N}_i^{\text{in}}}^d) \\ &\quad + \max_{x \in \mathcal{U}_i} g(x | X_{\mathcal{N}_i^{\text{in}}}^d) - \max_{x \in \mathcal{U}_i} \tilde{g}_i(x | X_{\mathcal{N}_i^{\text{in}}}^d) \\ &\leq \tilde{g}_i(x_i^d | X_{\mathcal{N}_i^{\text{in}}}^d) - g(x_i^d | X_{\mathcal{N}_i^{\text{in}}}^d) \\ &\quad + g(\hat{x} | X_{\mathcal{N}_i^{\text{in}}}^d) - \tilde{g}_i(\hat{x} | X_{\mathcal{N}_i^{\text{in}}}^d), \\ &\quad \text{for } \hat{x} \in \arg \max g(\hat{x} | X_{\mathcal{N}_i^{\text{in}}}^d) \\ &\leq \gamma_i^{\text{obj}}. \end{aligned} \quad (39)$$

The equality in (39) follows by expanding and rearranging the costs (13) on the left-hand-side. The first inequality swaps the subtracted maximum for the approximate marginal gain \tilde{g}_i at the point of the first maximum. Then, the second inequality uses the definition of the objective cost (15) (maximum over- and under- approximation) to bound the two differences.

Then, the expression for the costs in (17)

$$\gamma_i^{\text{gen}}(g, x_i^{\text{d}}, X_{1:i-1}^{\text{d}}) \leq \gamma_i^{\text{obj}} + \gamma_i^{\text{plan}} + \gamma_i^{\text{dist}} \quad (40)$$

follows by substituting the prior three equations into each other: (39) into (38) and the result into (37). Finally, substituting this inequality (40) into (19) from Lemma 2 (on the suboptimality of general assignments) yields the desired bound (17) which completes the first part of this proof.

2) *Proof of Theorem 1, part 2* (18): The second part of Theorem 1 (18) follows by referring to definition of γ_i^{dist} in (14), applying the chain rule (29), and substituting the definitions of the weights (26) and (27) in turn:

$$\gamma_i^{\text{dist}} = -g(x_i^{\text{d}}; X_{\mathcal{N}_i^{\text{in}}}^{\text{d}} | X_{\mathcal{N}_i^{\text{in}}}^{\text{d}}) \quad (41)$$

$$= - \sum_{j \in \mathcal{N}_i^{\text{in}}} g(x_i^{\text{d}}; x_j^{\text{d}} | X_{\mathcal{N}_i^{\text{in}}}^{\text{d}}, X_{\mathcal{N}_i^{\text{in}} \cap \{1:j-1\}}^{\text{d}}) \quad (42)$$

$$\leq \sum_{j \in \mathcal{N}_i^{\text{in}}} \mathcal{W}(i, j) \leq \sum_{j \in \mathcal{N}_i^{\text{in}}} \widehat{\mathcal{W}}(i, j). \quad (43)$$

Then, (18) follows by summing over \mathcal{R} . This completes this second and last part of the proof of Theorem 1. ■

APPENDIX IV

ANALYSIS FOR SCALING TO LARGE NUMBERS OF ROBOTS

The analysis in this section establishes sufficient conditions for the cost of distributed planning γ^{dist} (14) for each robot to be constant (in expectation) for planners with a fixed number of sequential steps, independent of the number of robots. Afterward, we discuss how this analysis relates to the design and analysis of target tracking systems.

A. Bounding expected inter-agent redundancy

Consider a distribution of robots and targets on \mathbb{R}^n with at most α robots and β targets on average per unit volume. Then, assume that the channel capacities (10) between each robot $i \in \mathcal{R}$ and target $j \in \mathcal{T}$ satisfy a non-increasing upper bound $\phi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ (possibly in expectation) so that $C_{i,j} \leq \phi(\|\mathbf{p}_i^{\text{r}} - \mathbf{p}_j^{\text{t}}\|_2)$ where \mathbf{p}_i^{r} and \mathbf{p}_j^{t} are the robot position and target *mean* position in \mathbb{R}^n .

Now, consider the expectation of the total weight associated with robot $i \in \mathcal{R}$ considering only robots and targets on an n -ball with radius $R/2$ centered around \mathbf{p}_i^{r} which we write as $\mathbb{E}_{B_{R/2}}[\sum_{j \in \mathcal{R} \setminus \{i\}} \widehat{\mathcal{W}}(i, j)]$. Consider also the expectation for targets distributed within a radius R around i and robots on balls with the same radius centered on each target. The intersection of these balls is a ball around i with radius $R/2$ so that the latter expectation produces an upper bound on the former. Given the expression for $\widehat{\mathcal{W}}$ in terms of channel capacities (27), and designating the zero-centered ball with

radius R as B_R we can write this inequality as:

$$\begin{aligned} & \mathbb{E}_{B_{R/2}} \left[\sum_{j \in \mathcal{R} \setminus \{i\}} \widehat{\mathcal{W}}(i, j) \right] \\ & \leq \int_{B_R} \int_{B_R} \alpha \beta \min(\phi(\|\mathbf{x}\|_2), \phi(\|\mathbf{y}\|_2)) \, d\mathbf{x} \, d\mathbf{y}. \end{aligned} \quad (44)$$

By integrating over the surface of each ball (each an $(n-1)$ -sphere with surface area S_{n-1})

$$= \alpha \beta \int_0^R \int_0^R S_{n-1}^2 r_1^{n-1} r_2^{n-1} \min(\phi(r_1), \phi(r_2)) \, dr_1 \, dr_2. \quad (45)$$

Given that ϕ is non-increasing, separating the minimum produces:

$$\begin{aligned} & = \alpha \beta S_{n-1}^2 \left(\int_0^R \int_0^{r_2} r_1^{n-1} r_2^{n-1} \phi(r_2) \, dr_1 \, dr_2 \right. \\ & \quad \left. + \int_0^R \int_{r_2}^R r_1^{n-1} r_2^{n-1} \phi(r_1) \, dr_1 \, dr_2 \right), \end{aligned} \quad (46)$$

and by swapping the bounds of the second integral, combining, and evaluating the inner integral, we get:

$$\begin{aligned} & = \alpha \beta S_{n-1}^2 \left(\int_0^R \int_0^{r_2} r_1^{n-1} r_2^{n-1} \phi(r_2) \, dr_1 \, dr_2 \right. \\ & \quad \left. + \int_0^R \int_0^{r_1} r_1^{n-1} r_2^{n-1} \phi(r_1) \, dr_2 \, dr_1 \right) \end{aligned} \quad (47)$$

$$= 2\alpha \beta S_{n-1}^2 \int_0^R \int_0^{r_1} r_1^{n-1} r_2^{n-1} \phi(r_1) \, dr_2 \, dr_1 \quad (48)$$

$$= \frac{2\alpha \beta S_{n-1}^2}{n} \int_0^R r_1^{2n-1} \phi(r_1) \, dr_1. \quad (49)$$

The above integral (49) converges in the limit if $\phi \in O(1/x^{2n+\epsilon})$. Most relevantly, for a plane, this condition comes to $\phi \in O(1/x^{4+\epsilon})$.¹³

B. Scaling and sensor models

The sensitivity to how quickly interactions between robots and targets fall off motivates attention to sensing design and modeling to prevent distributed planning from performing poorly for large numbers of robots or else requiring additional computation time. For example, considering additive Gaussian noise with a standard deviation proportional to distance (as is common in range sensing models [4]) mutual information does not fall off quickly enough as is evident from the capacity of Gaussian channels [28, Chap. 9].

At the same time, robots in realizable systems cannot obtain and process observations of unbounded numbers of targets, and features such as a maximum sensor range (as we use in the results) can model such limits. Still, the observation of whether a target is within range provides

¹³This requirement on interactions between *robots and targets* is stricter than the equivalent one *between robots* [16, Sec. VI.C] as interactions between robots must decrease as $O(1/x^{n+\epsilon})$.

some information. Introducing a narrow tails assumption on the target filters—such as that the *tails approach zero exponentially*—in combination with a maximum sensing range

ensures that ϕ decreases sufficiently quickly. However, due to this sensitivity to the tails, the scaling behavior (49) may be difficult to estimate, even when known to be bounded.