



Brief paper

Optimal composition of heterogeneous multi-agent teams for coverage problems with performance bound guarantees[☆]Chuangchuang Sun^{a,*}, Shirantha Welikala^b, Christos G. Cassandras^b^a Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA^b Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA 02446, USA

ARTICLE INFO

Article history:

Received 9 September 2019

Received in revised form 3 January 2020

Accepted 19 March 2020

Available online 9 April 2020

Keywords:

Multi-agent systems

Optimization

Cooperative control

ABSTRACT

We consider the problem of determining the optimal composition of a heterogeneous multi-agent team for coverage problems by including costs associated with different agents and subject to an upper bound on the maximal allowable number of agents. We formulate a resource allocation problem without introducing additional non-convexities to the original problem. We develop a distributed Projected Gradient Ascent (PGA) algorithm to solve the optimal team composition problem. To deal with non-convexity, we initialize the algorithm using a greedy method and exploit the submodularity and curvature properties of the coverage objective function to derive novel tighter performance bound guarantees on the optimization problem solution. Numerical examples are included to validate the effectiveness of this approach in diverse mission space configurations and different heterogeneous multi-agent collections. Comparative results obtained using a commercial mixed-integer nonlinear programming problem solver demonstrate both the accuracy and computational efficiency of the distributed PGA algorithm.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Cooperative multi-agent systems are pervasive in a number of applications, including but not limited to, surveillance (Castanedo, García, Patricio, & Molina, 2010), search and rescue missions (Luo, Espinosa, Pranantha, & De Gloria, 2011), consensus (Zheng, Ma, & Wang, 2018) and agriculture (Balmann, 2000). One of the most basic tasks such a system can perform that has seen a wide range of applications is *coverage*. The fundamental multi-agent optimal coverage problem has been extensively studied in the literature, e.g., Breitenmoser, Schwager, Metzger, Siegwart, and Rus (2010), Caicedo-Nunez and Zefran (2008), Cassandras and Li (2005) and Meguerdichian, Koushanfar, Potkonjak, and Srivastava (2001). In this problem, agents are deployed to “cover” as much of a given mission space as possible in the sense that the team aims to optimally jointly detect events of interest (e.g., data sources) that may randomly occur anywhere in this

space. The coverage performance is measured by an appropriate metric, which is normally defined as the joint event detection probability. The optimal coverage problem is particularly challenging due to the generally non-convex nature of this metric and of the mission space itself due to the presence of obstacles which act as constraints on the feasible agent locations that constitute a solution to the problem.

Thus far, the analysis of the optimal coverage problem has been carried out based on the assumption that there exists a fixed number N of agents to be deployed. However, this number is often limited by cost constraints, leading to a natural trade-off between coverage performance (which is normally monotonically increasing in N) and total system cost. In such a setting, an additional aspect of the problem is that of managing a set of *heterogeneous* agents: when agents fall into different classes characterized by different properties such as sensing capacity, range, attenuation rate, and cost, then the problem becomes one of determining the *optimal cooperative team composition* in terms of the number of agents selected from each class so as to optimize an appropriate metric capturing the performance-cost trade-off. Clearly, it is possible that a certain team composition can achieve the same coverage performance as another, but with a lower cost due to the heterogeneity of agents. The purpose of this paper is to address the optimal coverage problem in the presence of heterogeneous agents under cost constraints.

As mentioned above, the optimal coverage problem is already challenging due to its non-convex nature. Heuristic algorithms

[☆] This work is supported in part by NSF under grants ECCS-1509084, DMS-1664644, CNS-1645681, by AFOSR under grant FA9550-19-1-0158, by ARPA-E's NEXTCAR program under grant DE-AR0000796 and by the MathWorks. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Tongwen Chen under the direction of Editor Ian R. Petersen.

* Corresponding author.

E-mail addresses: ccsun1@mit.edu (C. Sun), shiran27@bu.edu (S. Welikala), cgc@bu.edu (C.G. Cassandras).

(e.g., genetic algorithms Davis, 1996), are often used and may lead to empirically near-global optimality, but they are prohibitively inefficient for on-line use. On the other hand, on-line algorithms sacrifice potential optimality to achieve efficiency; this includes distributed gradient-based algorithms (Cassandras & Li, 2005; Gusrialdi & Zeng, 2011; Zhong & Cassandras, 2011) and Voronoi-partition-based algorithms (Breitenmoser et al., 2010; Cortes, Martinez, Karatas, & Bullo, 2004; Gusrialdi, Hirche, Hatanaka, & Fujita, 2008) which lead to generally locally optimal solutions. Methods for efficiently escaping such local optima using a “boosting function” approach were proposed in Sun, Cassandras, and Gokbayrak (2014) and Welikala and Cassandras (2020), while a decentralized control law in Schwager, Bullo, Skelly, and Rus (2008) seeks a combination of optimal coverage and exploration.

A parallel effort to deal with the difficulty of finding a globally optimal solution for the basic coverage problem is by exploiting the *submodularity* properties of the coverage performance functions. This is accomplished in Sun, Cassandras, and Meng (2019) by using a greedy algorithm to initialize the state of the system (i.e., the locations of the agents), followed by a conventional gradient ascent technique to obtain an improved (still locally optimal) solution. Due to submodularity, the ratio f^G/f^* , where f^G and f^* correspond to the objective function values under a greedy solution and the globally optimal solution respectively, has a lower bound $L \leq f^G/f^*$ which is shown to be $L = 1/2$ in Fisher, Nemhauser, and Wolsey (1978). When the objective function f is *monotone submodular* (which applies to coverage metrics), then it has been shown that $L = (1 - \frac{1}{e})$ (Nemhauser, Wolsey, & Fisher, 1978) and becomes $L = (1 - (1 - \frac{1}{N})^N)$ when the allowable maximum number of agents is constrained to N . Recent work (Conforti & Cornuéjols, 1984; Liu, Chong, & Pezeszki, 2018; Wang, Moran, Wang, & Pan, 2016) has further improved these performance bounds by exploiting the *curvature* properties of the specific objective function. By using these improved bounds, the solutions to a variety of optimal coverage problems in Sun, Cassandras et al. (2019) have been shown to often approach $L = 1$, i.e., to yield almost globally optimal solutions.

Our contributions in this paper are threefold. First, we formulate the problem of determining an optimal team composition under a heterogeneous set of agents as a resource allocation problem without introducing additional non-convexity features to it. In our problem formulation, instead of imposing a hard cardinality constraint, an l_1 norm penalty in the objective function is employed to induce sparsity and prevent any new non-convexity from being introduced. Secondly, for the coverage component of the objective function (i.e., without the aforementioned penalty term), a greedy algorithm is used and two new improved performance bounds are derived based on the concepts of partial curvature (Liu et al., 2018), total curvature, and greedy curvature (Conforti & Cornuéjols, 1984). Finally, we propose a distributed projected gradient ascent algorithm to solve the overall optimal team composition problem. The key to this algorithm is the proper selection of an *initial condition* which is characterized by a provable lower bound. Thus, we first use a greedy method to generate a candidate solution to the underlying coverage component of the problem which always contains all the available agents. This is used as the initial condition for a distributed projected gradient ascent scheme whose final solution recovers both the integer and real variables associated with the problem which respectively define the optimal team composition and the optimal agent locations.

A crucial difference in this work compared to Sun, Cassandras et al. (2019) and Zhong and Cassandras (2011) is that we do not assume that a given number of agents is to be deployed; rather, we seek to determine the number of agents (subject to an upper bound constraint) and optimal team composition (not only

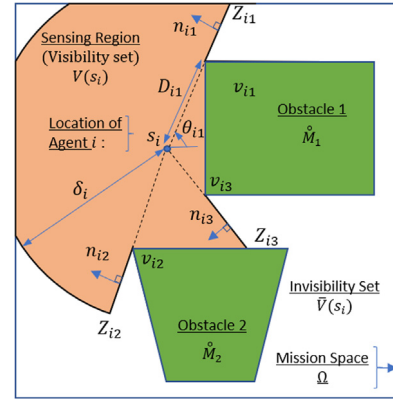


Fig. 1. Mission space with obstacles.

the optimal agent locations), which is a combinatorial NP-hard problem. In addition, two new tighter performance bounds are derived compared to those in Sun, Cassandras et al. (2019).

The rest of the paper is organized as follows. The optimization problem for determining the optimal team composition is formulated in Section 2. Then, to obtain a good initial condition to solve this optimization problem, a greedy algorithm is presented in Section 3, along with some performance bound guarantees. Subsequently, to completely solve the formulated optimization problem, a distributed projected gradient ascent process is proposed in Section 4, along with some theoretical results regarding the nature of its terminal solution. Numerical results are included in Section 5 to validate the effectiveness of the proposed solution technique. Finally, Section 6 concludes the paper.

2. Problem formulation

We begin with a brief review of the multi-agent coverage problem (see Caicedo-Nunez & Zefran, 2008; Cortes et al., 2004; Zhong & Cassandras, 2011). The mission space $\Omega \subseteq \mathbb{R}^2$ is modeled as a convex compact polygon. For non-convex polygons Ω_1 , such as the self-intersecting ones, we make Ω the convex hull of Ω_1 , while $\Omega \setminus \Omega_1$ defines obstacles that agents have to avoid. Let $R(x) : \mathbb{R}^2 \rightarrow \mathbb{R}$ be an event density function such that $R(x) \geq 0, \forall x \in \Omega$ and $\int_{\Omega} R(x) dx < \infty$ such that $R(x)$ represents the relative importance of a point $x \in \Omega$. Obstacles present in the mission space can both limit the movement of agents and interfere with their sensing capacities. Such obstacles are modeled as non-intersecting polygons $M_1, \dots, M_m$ and their interiors are forbidden regions for the agents. As a result, the feasible (safety) region is $F = \Omega \setminus (\dot{M}_1 \cup \dots \cup \dot{M}_m)$, where $\dot{M}$ is the interior of M .

With N as the maximum possible number of agents, we have $\mathbf{s} = [s_1^T, \dots, s_N^T]^T \in \mathbb{R}^{N \times 2}$ denoting the locations of the N agents with each $s_i \in \mathbb{R}^2, \forall i = 1, \dots, N$. Then, the following sensing model is adopted. For any point $x \in \Omega$ and a certain agent at s_i , there are two issues affecting if the agent can detect an event occurring at x . First, the agent is characterized by a sensing region defined as $\Omega_i = \{x | \|x - s_i\| \leq \delta_i\}$, where δ_i is the sensing range. Secondly, obstacles prevent a signal at x from reaching s_i . This is described by the condition $\eta s_i + (1 - \eta)x \in F, \eta \in [0, 1]$, i.e., the segment connecting x and s_i must be contained in the feasible region. Then, the visibility set of s_i is defined as $V(s_i) = \Omega_i \cap \{x | \eta s_i + (1 - \eta)x \in F\}$ and the invisibility set $\bar{V}(s_i)$ is the complement of $V(s_i)$ in F , i.e., $\bar{V}(s_i) = F \setminus V(s_i)$. An illustration of $V(s_i)$ is shown in Fig. 1. The probability that agent i detects an event at x in an unconstrained environment is given by

$$p_i(x, s_i) = p_{i0} e^{-\lambda_i \|x - s_i\|} \quad (1)$$

where $p_{i0} \in (0, 1]$ is the agent's sensing capacity and $\lambda_i > 0$ is a sensing decay (attenuation) factor. Different $p_i(x, s_i)$ specified by p_{i0} , δ_i and λ_i will lead to a heterogeneous multi-agent system. In a mission space with constraints, the agent's detection probability becomes:

$$\hat{p}_i(x, s_i) = \begin{cases} p_i(x, s_i) & \text{if } x \in V(s_i), \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Finally, assuming detection independence among the N agents, the joint detection probability of an event at x is given by $\hat{P}(x, \mathbf{s}) = 1 - \prod_{i=1}^N (1 - \hat{p}_i(x, s_i))$. As formulated in Zhong and Cassandras (2011), the optimal multi-agent coverage problem is

$$\max_{\mathbf{s}} H(\mathbf{s}) = \int_{\Omega} R(x) \hat{P}(x, \mathbf{s}) dx \quad (3)$$

s.t. $s_i \in F$, $i = 1, \dots, N$,

where the number of agents N is a predetermined constant. When N is in fact an additional decision variable constrained by the cost of agents, we proceed by capturing the trade-off between improved performance, which monotonically increases with N , and agent cost as follows. Letting N be the *maximum possible number* of agents to consider, we formulate a resource (sensing capacity) allocation problem:

$$\max_{\mathbf{s}, \mathbf{t}} H(\mathbf{s}, \mathbf{t}) = \int_{\Omega} R(x) P(x, \mathbf{s}, \mathbf{t}) dx - \beta \sum_{i=1}^N t_i \quad (4)$$

s.t. $s_i \in F$, $t_i \in \{0, 1\}$, $i = 1, \dots, N$,

with $P(x, \mathbf{s}, \mathbf{t}) = 1 - \prod_{i=1}^N (1 - t_i \hat{p}_i(x, s_i))$. In (4), $\mathbf{t} = [t_1, t_2, \dots, t_N]^T$ and t_i is a binary decision variable associated with agent i . The term $\beta \sum_{i=1}^N t_i$ denotes the cost of deploying N agents, where $\beta \geq 0$ is a weight capturing the cost of each agent (assumed to be the same in this formulation). In order to ensure a properly normalized objective function, β must be selected to be consistent with the following convex combination of objectives: $\hat{H}(\mathbf{s}, \mathbf{t}) = w_1 \frac{1}{\int_{\Omega} R(x) dx} \int_{\Omega} R(x) P(x, \mathbf{s}) dx - (1 - w_1) \frac{1}{N} \sum_{i=1}^N t_i$, where $w_1 \in (0, 1]$ (resp. $1 - w_1$) and $\int_{\Omega} R(x) dx$ (resp. N) are weights associated with the coverage performance metric (resp. cost function). Observing that each component above is properly normalized in $[0, 1]$, we can adopt (4) as long as β is selected so that $\beta = \frac{1 - w_1}{w_1} \frac{\int_{\Omega} R(x) dx}{N}$. Note that with $t_i \in \{0, 1\}$, the agent heterogeneity in $\hat{p}_i$ (which depends on the values of p_{i0} and λ_i in (1) and on the sensing range δ_i) is not included in the formulation (4). In order to capture this aspect of the problem, we relax the binary nature of t_i by allowing it to be a continuous variable $t_i \in [0, 1]$. We then rewrite the detection probability in (1) as $t_i p_{i0} e^{-\lambda_i \|x - s_i\|}$ so that t_i acts as a discount factor for the sensing capacity p_{i0} . Accordingly, (2) is modified to

$$\bar{p}_i(x, s_i, t_i) = \begin{cases} t_i p_{i0} e^{-\lambda_i \|x - s_i\|} & \text{if } x \in V(s_i), \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

and the definition of the joint detection probability in becomes $\bar{P}(x, \mathbf{s}, \mathbf{t}) = 1 - \prod_{i=1}^N (1 - \bar{p}_i(x, s_i, t_i))$. With $\bar{P}(x, \mathbf{s}, \mathbf{t})$ as defined above, we now extend (4) to

$$\max_{\mathbf{s}, \mathbf{t}} \int_{\Omega} R(x) \bar{P}(x, \mathbf{s}, \mathbf{t}) dx - \beta \sum_{i=1}^N t_i \quad (6)$$

s.t. $s_i \in F$, $t_i \in [0, 1]$, $i = 1, \dots, N$.

However, this formulation still does not capture the fact that agents with different sensing parameter values p_{i0} , δ_i and λ_i have different costs. Therefore, let $\gamma_i(p_{i0}, \lambda_i, \delta_i)$ denote the cost of agent i and let us still keep β as a weight indicating the overall importance of cost relative to the coverage performance

expressed by the first term in the objective function. Omitting the dependence of γ_i on the sensing parameters, we now formulate the problem:

$$\max_{\mathbf{s}, \mathbf{t}} H(\mathbf{s}, \mathbf{t}) = \int_{\Omega} R(x) \bar{P}(x, \mathbf{s}, \mathbf{t}) dx - \beta \sum_{i=1}^N \gamma_i t_i \quad (7)$$

s.t. $s_i \in F$, $t_i \in [0, 1]$, $i = 1, \dots, N$.

Clearly, heterogeneity here is captured in two ways: first, by imposing a different cost γ_i to each agent and second by associating a different sensing capacity $t_i p_{i0}$ in (5) to each agent, assuming that such capacity is adjustable. More importantly, while the binary constraint in (4) is removed, the l_1 norm used in (7) is a regularization term which is well known to induce sparsity (Tibshirani, 1996). The implication is that solutions of this problem will tend to include values $t_i = 0$ for several agents in seeking cost-effective team compositions. This is both theoretically proven in Theorem 2 and experimentally validated in Section 5.

As in the case of (4), the objective function in (7) needs to be properly normalized. To accomplish this while also providing a physical interpretation to the cost coefficients γ_i , recall that $\Omega_i = \{x | \|x - s_i\| \leq \delta_i\}$ represents the sensing region of agent i and define the *sensing capability* of this agent as $\kappa_i = \int_{\Omega_i} \bar{p}_i(x, s_i) dx$, where $s_i \in \mathbb{R}^2$ can be any point in the boundless and obstacle-free space, hence κ_i is independent of s_i ; it depends only on the sensing parameters p_{i0} , λ_i , and δ_i . In fact, for the exponential sensing function given in (1), a closed-form expression for κ_i can be obtained as $\kappa_i = \frac{2\pi p_{i0}}{\lambda_i^2} [1 - (1 + \lambda_i \delta_i) e^{-\lambda_i \delta_i}]$. Now, assuming

the cost γ_i associated with agent i is proportional to its sensing capability κ_i , we write $\gamma_i = w_{2i} \kappa_i$, where $w_{2i} \in (0, 1]$ is a prespecified *agent cost weight*. Finally, we update the definition of the normalization factor β in (4) as follows:

$$\beta = \frac{1 - w_1}{w_1} \frac{\int_{\Omega} R(x) dx}{\sum_{i=1}^N \gamma_i}. \quad (8)$$

With that, we can compute all the parameters/coefficients in (7), when the agent sensing capabilities and weights are given.

3. Greedy algorithm and submodularity theory for coverage problems

In order to obtain an initial solution to problem (7), we first consider problem (3) where the objective is limited to maximizing the coverage using all the available agents. We adopt the generic greedy method proposed in Sun, Cassandras et al. (2019) and seek to improve the performance bounds provided in Sun, Cassandras et al. (2019) by exploiting the curvature concepts proposed in Conforti and Cornuéjols (1984) and Liu et al. (2018).

In order to take advantage of the submodular structure of $H(\mathbf{s})$ in (3), we first uniformly discretize the continuous feasible space F to form a *ground-set* $F^D = \{x_1, x_2, \dots, x_n\}$ with each $x_i \in F$. These x_i values can be thought of as feasible points where an agent can be placed. Note that the cardinality $|F^D|$ of the ground-set is $|F^D| = n$. Next, a *set-variable* is defined as $S = \{s_1, s_2, \dots\}$ to represent the initial placement for each agent. And a *set-constraint* $S \in \mathcal{S}$, where $\mathcal{S} = \{A : A \subseteq F^D, |A| \leq N\}$ is introduced to convey that each $s_i \in S$ should be chosen from F^D and agent team size $|S|$ should be constrained to N . Typically, a set-constraint of this form is called a *uniform matroid constraint of rank N* where the pair $\mathcal{M} = (F^D, \mathcal{S})$ is known as a *uniform matroid*. Finally, we approximate the coverage objective function $H(\mathbf{s})$ in (3) by a set-function $H(S)$, where $H : \mathcal{S} \rightarrow \mathbb{R}$ and

$$H(S) = \int_{\Omega} R(x) \left(1 - \prod_{s_i \in S} [1 - \hat{p}_i(x, s_i)]\right) dx. \quad (9)$$

Therefore, $H(S)$ now represents the coverage objective value achieved by the agent placement defined by the set-variable S . In this new framework, a set-function version of the original coverage problem in (3) can be written as

$$\max_S H(S) \quad \text{s.t. } S \in \mathcal{S}. \quad (10)$$

Greedy algorithm. Due to the combinatorial search space size, an exact solution to (10) is challenging to obtain. However, a candidate solution can be obtained using a simple greedy algorithm (see Alg. 1) and is referred to as a *greedy solution*.

The marginal gain in the coverage objective due to adding a new agent at $x_i \in F^D$ to an existing agent set A is defined as $\Delta H(x_i|A) \triangleq H(A \cup \{x_i\}) - H(A)$. It can be shown that

$$\Delta H(x_i|A) = \int_F R(x) p_i(x, s_i) \prod_{s_j \in A} [1 - \hat{p}_j(x, s_j)] dx, \quad (11)$$

using (9). As the next step, properties of the problem (10) are used to derive several bounds to quantify how close the greedy solution is to the globally optimal solution.

Performance bounds. The *performance ratio* of the greedy solution S^G (of (10) given by Alg. 1) is defined as $\frac{H(S^G)}{H(S^*)}$ where S^* is the globally optimal solution (of (10), generally unknown). A *performance bound* L is a theoretically imposed lower bound to the performance ratio. Therefore,

$$L \leq \frac{H(S^G)}{H(S^*)} \leq 1. \quad (12)$$

It was proven in Sun, Cassandras et al. (2019) that the set-function $H(S)$ has two important properties: *submodularity* and *monotonicity*. Hence, following the seminal paper (Nemhauser et al., 1978), the greedy solution of (10) is characterized by $L = L_C$, referred to as the *conventional* performance bound:

$$L_C = (1 - (1 - \frac{1}{N})^N). \quad (13)$$

3.1. Incorporating curvature information

For the class of coverage problems, it is shown in Sun, Cassandras et al. (2019) that tighter performance bounds (i.e., performance bounds which are closer to 1 than L_C) can be obtained using the *curvature* information of the objective function $H(S)$. Typically, any measure of *curvature* of a set function $f(A)$ provides additional information about the nature of its growth when new elements are added to the set-variable A . For example, the marginal gain of a coverage objective set-function $H(S)$ (represented by $\Delta H(\cdot|S)$), can drastically drop when elements are added to the set S . Due to this reason, curvature information can yield vital information about the effectiveness of greedy methods (more details can be found in Sun, Welikala and Cassandras, 2019).

Total curvature. The concept of *total curvature* for generic submodular monotone set-functions was introduced in Conforti and Cornuéjols (1984). For the coverage problem in (10), it is denoted by α_T ,

$$\alpha_T = \max_{x_i: x_i \in F^D} \left[1 - \frac{\Delta H(x_i|F^D \setminus x_i)}{\Delta H(x_i|\emptyset)} \right]. \quad (14)$$

Here $\emptyset$ and “ $\setminus$ ” denote the empty set and the set-subtraction operation respectively. Using (11), (9), and the knowledge of F^D , the total curvature α_T (14) of the set-function $H(S)$ can be explicitly evaluated. In Conforti and Cornuéjols (1984), it is shown that when maximizing a submodular monotone set function with a total curvature α_T , the greedy solution will follow the performance

Algorithm 1 Greedy Method for Solving (10)

```

1: Inputs:  $N, F^D, \mathcal{S}$ ; Outputs: Greedy solution  $S^G$ .
2:  $S := \emptyset$ ;  $i := 1$ ;
3: while  $i \leq N$  do
4:    $s_i := \arg \max_{x_i: (S \cup \{x_i\}) \in \mathcal{S}} (\Delta H(x_i|S))$ ;
5:    $S := S \cup \{s_i\}$ ;
6: end while
7:  $S^G := S$ ; Return;

```

bound $L = L_T = \frac{1}{\alpha_T} [1 - (\frac{N - \alpha_T}{N})^N]$. This total curvature measure has been used in Sun, Cassandras et al. (2019) to establish better performance bounds compared to the conventional bound L_C in the context of the coverage control problem in (10). Next, we propose another curvature concept to obtain even tighter performance bounds than L_T .

Partial curvature. In Liu et al. (2018), the concept of *partial curvature* is proposed for submodular monotone set functions which are defined under uniform matroid constraints. Adopting this new concept, the partial curvature measure associated with the coverage objective set-function $H(S)$ can be expressed as α_P where

$$\alpha_P = \max_{(A, x_i): x_i \in A \in \mathcal{S}} \left[1 - \frac{\Delta H(x_i|A \setminus x_i)}{\Delta H(x_i|\emptyset)} \right]. \quad (15)$$

As discussed in Liu et al. (2018), the partial curvature delivers a better characterization of the monotonicity of any generic set-function compared to the total curvature. This improvement is due to the fact that only the information obtained from the domain of the considered set-function is used, which can be considerably smaller due to the uniform matroid constraint. The importance of the partial curvature concept in the context of our coverage problem can be explained as follows. For coverage problems, evaluating $H(F^D)$ so as to compute the total curvature in (14) and then to impose the performance bound L_T is problematic because the domain of $H(\cdot)$ in the original optimization problem (10) is actually limited to size N sets (i.e., by the constraint $S \in \mathcal{S}$). This issue is critical when we consider heterogeneous agents (in terms of sensing capabilities) and a finite set of agents at our disposal to achieve the maximum coverage. In such situations, $H(F^D)$ is ill-defined and, therefore, the total curvature and the respective performance bound L_T cannot be evaluated. However, the definition of the partial curvature in (15) will still hold as it only requires evaluations of $H(\cdot)$ over the same domain (i.e., $S \in \mathcal{S}$).

Using (11), (9) and the knowledge of $\mathcal{S}$, the partial curvature α_P in (15) can be computed for the coverage problem. The corresponding performance bound is $L = L_P = \frac{1}{\alpha_P} [1 - (\frac{N - \alpha_P}{N})^N]$.

Greedy curvature. This curvature concept is proposed in Conforti and Cornuéjols (1984) as an on-line method of estimating a performance bound. The resulting performance bound depends on the greedy solution S^G itself. Note that the performance bounds discussed thus far are not dependent on the obtained greedy solution but only the objective function parameters (such as λ_i , δ_i for all i) and N , as well as the feasible space F^D . If the greedy algorithm given in Alg. 1 produces the solution sets $\emptyset = S^0 \subseteq S^1 \subseteq S^2 \subseteq \dots \subseteq S^N$ during the course of execution (where $S^N = S^G$), then, the greedy curvature metric α_G is

$$\alpha_G = \max_{0 \leq i \leq N-1} \left[\max_{x_j \in F^i} \left(1 - \frac{\Delta H(x_j|S^i)}{\Delta H(x_j|\emptyset)} \right) \right], \quad (16)$$

where $F^i = \{x_j : x_j \in F^D \setminus S^i, (S^i \cup \{x_j\}) \in \mathcal{S}\}$ is the set of valid points considered for the placement of the $(i+1)$ th agent during

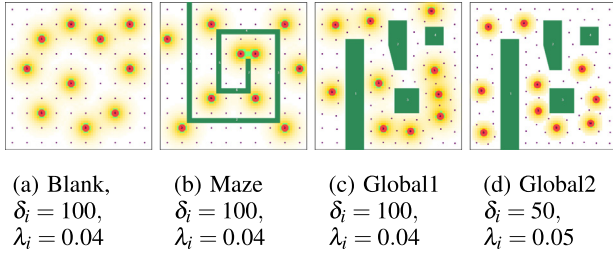


Fig. 2. Different problem settings and their greedy solutions for $N = 10$. Red dots are greedy agent locations, black dots represent the ground set. Darker colored areas have greater coverage, and green colored shapes are obstacles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the $(i + 1)$ th greedy iteration. Therefore, α_G can be computed in parallel with the greedy method (without performing any additional computations) unlike the previously discussed two cases. The corresponding performance bound is $L = L_G = 1 - \alpha_G(1 - \frac{1}{N})$.

The main idea behind the greedy curvature concept is that the solution sets generated during the greedy algorithm itself can be used to characterize the monotonicity of the considered set-function and then to establish a performance bound based on that information. Therefore, similar to the feasibility of using the total curvature-based performance bound L_T for a heterogeneous set of agents, the definition of the greedy curvature measure in (16) and the performance bound L_G will still hold in such cases.

The overall performance bound L . Considering all the performance bounds L_C, L_T, L_P and L_G defined above, an overall performance bound L satisfying (12) can be established as

$$L = \max \{L_C, L_T, L_P, L_G\}. \quad (17)$$

Generally, $L_C \leq L_T \leq L_P$ (Liu et al., 2018). Also, when heterogeneous agents are involved, $L_C = 0.5$ and L_T and L_G are undefined.

3.2. Numerical results for greedy method

We now investigate the behavior of the proposed partial curvature and greedy curvature-based performance bounds L_P, L_G compared to the total curvature and conventional performance bounds L_C, L_T . Four different representative problem settings were considered as shown in Fig. 2. Under each of these settings, L_P, L_G, L_T and L_C were evaluated for different values of the total allowable number of agents N . From the obtained results shown in Fig. 3, it is evident that the proposed use of partial curvature always delivers better bounds than the total curvature approach (Conforti & Cornuéjols, 1984). Similarly, the proposed use of greedy curvature provides better bounds than the total curvature approach (Conforti & Cornuéjols, 1984) when N takes moderate values (i.e., N is around 2 – 20). Moreover, L_G is useful for computation-limited settings, as it does not require any additional computations compared to evaluating L_T or L_P .

As pointed out earlier, the performance bound L_T is ill-defined when considering heterogeneous agents. To avoid this problem, the experiments reported above were limited to a homogeneous set of agents. However, it should be emphasized that the definitions of the proposed performance bounds L_P and L_G are robust to agent heterogeneity, the situation considered in Section 5. Therefore, in such heterogeneous situations, using L_P and/or L_G will be the only way to obtain an improved performance bound compared to the conventional bound L_C . Note that in such situations, the greedy algorithm given will require an additional inner loop to determine the optimal type of the agent to be

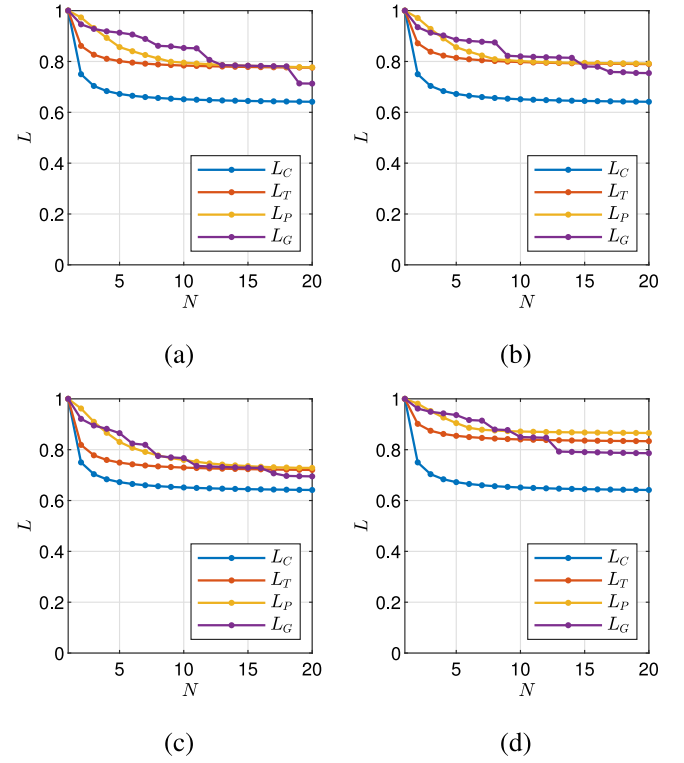


Fig. 3. Performance bounds (as a function of N): (i) Conventional L_C , (ii) Total curvature L_T , (iii) Partial curvature L_P , and (iv) Greedy curvature L_G , for the four problem settings in Fig. 2.

deployed at each main greedy iteration. We conclude this section by reminding the reader that the greedy process detailed above is needed so as to generate an initial condition to the main optimization problem in (7).

4. A gradient based algorithm for heterogeneous multi-agent coverage problems

The greedy algorithm (Alg. 1) is limited to discrete environments and a fixed predetermined agent number. Its value in solving the actual problem of interest in (7) is twofold: (i) Provide a reasonable initial condition for a gradient-based algorithm used to solve (7) which can significantly overcome the local-optimality limitation of such an algorithm, and (ii) Provide a lower bound for the ultimate coverage performance we obtain. In this section, we propose a distributed gradient-based algorithm similar to that in Zhong and Cassandras (2011) aimed at solving (7). We first derive the derivatives of the objective function $H(s, t)$ with regard to the variables (s, t) for the gradient ascent update. Setting $s_i = (s_{ix}, s_{iy})$, we begin with $\frac{\partial H(s, t)}{\partial s_{ix}}$ whose derivation was given in (Zhong & Cassandras, 2011):

$$\frac{\partial H(s, t)}{\partial s_{ix}} = \int_{V(s_i)} R(x) \Phi_i(x) \frac{\partial \bar{p}_i(x, s_i, t_i)}{\partial s_{ix}} dx + \sum_{j \in I_i} \left[\text{sgn}(n_{ijx}) \frac{\sin(\theta_{ij})}{D_{ij}} \int_0^{z_{ij}} R(\rho(r)) \Phi_i(\rho(r)) \bar{p}_i(\rho(r), s_i, t_i) r dr \right], \quad (18)$$

where $\Phi_i(x) = \prod_{k \in B_i} [1 - \bar{p}_k(x, s_k, t_k)]$, $D_{ij} = \|v_{ij} - s_i\|$, $\frac{\partial \bar{p}_i(x, s_i, t_i)}{\partial s_{ix}} = -\lambda_i \bar{p}_i(x, s_i, t_i) \frac{(s_i - x)_x}{\|s_i - x\|}$, and $\rho(r) = \rho_{ij}(r) = \frac{v_{ij} - s_i}{D_{ij}} \cdot r + v_{ij}$. In (18), $\text{sgn}(\cdot)$ represents the signum function and the subscript x is used to represent the x -component of a two dimensional vector. The second term in (18) is due to the linear shaped boundary

segments of the sensing region $V(s_i)$ formed due to the obstacle vertices $v_{ij} \in V(s_i)$. Such linear segments are lumped into a set $\Gamma_i = \{\Gamma_{i1}, \Gamma_{i2}, \dots\}$ where each linear segment Γ_{ij} can be characterized by four parameters: (i) end point Z_{ij} , (ii) angle θ_{ij} , (iii) obstacle vertex v_{ij} , and, (iv) unit normal direction n_{ij} . Therefore, Γ_{ij} can be thought of as a four-tuple $\Gamma_{ij} = (Z_{ij}, \theta_{ij}, v_{ij}, n_{ij})$. All these geometric parameters (for a generic setting) are illustrated in Fig. 2. Note that we assume: (i) obstacles are polygonal, and, (ii) sensing power at the edge of the sensing region is negligible. More detailed definitions and derivations are omitted for brevity, and interested readers are referred to Zhong and Cassandras (2011). A similar expression can be obtained for $\frac{\partial H(s, t)}{\partial s_{ij}}$. As detailed in Zhong and Cassandras (2011), the agent locations are assumed not to coincide with a reflex vertex, a polygonal inflection, or a bi-tangent where $H(s, t)$ is not differentiable (if such points have to be taken into consideration, then a subgradient can be used as an alternative to the gradient).

Additionally, the derivative $\frac{\partial H(s, t)}{\partial t_i}$ is obtained as follows:

$$\frac{\partial H(s, t)}{\partial t_i} = \int_{V(s_i)} R(x) \Phi_i(x) p_i(x, s_i) dx - \beta \gamma_i. \quad (19)$$

Here, the integration and differentiation are interchangeable since $P(x, s)$ is a continuous differentiable function of t_i . The first term in (19) represents a *local coverage* level achieved by the agent i in its sensing region $V(s_i)$. This local coverage level depends on the state variables (s, t) and is always positive. The second term in (19) represents a *local cost* resulting from agent cost γ_i and the normalization factor β . Note that this local cost value is a predefined positive constant for each agent. This multi-objective interpretation of (19) can be used to conclude that when the aforementioned local coverage level is less than the (fixed) local cost, the state variable t_i should be decreased to improve the global objective $H(s, t)$, and vice versa.

Algorithm 2 is a Projected Gradient Ascent (PGA) algorithm for solving (7) which utilizes the gradients derived in (18) and (19). As seen in Algorithm 2, a gradient ascent update is first implemented in (Eq. (20)), where $\eta_s^{(k)} > 0$, $\eta_t^{(k)} > 0$ are the step sizes chosen based on standard technical conditions (Bertsekas, 2016) (more application-specific details on the step size selection can be found in Welikala & Cassandras, 2020). Subsequently, the projection mechanisms are applied to guarantee the satisfaction of all constraints. The projection $\Pi_A(x)$ of $x \in \mathbb{R}^n$ onto a set $A \subseteq \mathbb{R}^n$ is formally defined as $\Pi_A(x) \triangleq \arg \min_{y \in A} \|x - y\|_2$. For $s_i \in F$, if the update direction (i.e., $\frac{\partial H(s, t)}{\partial s_i}$) is pointing directly into an obstacle's boundaries, then the update direction is projected onto the boundary itself and thus prevents violation of the obstacle constraint. As for the bound constraint for t_i , a projection onto the convex set $[0, 1]$ is simply a truncation.

Coverage performance of the PGA solution s^{PGA} . For the initialization of the PGA algorithm given in Alg. 2, we use the greedy solution $s^{(0)} = [S^G]$ obtained from Alg. 1 using: (i) The pre-specified discretized feasible space F^D , and, (ii) The complete set of agents (all N of them). The overall performance bound obtained using (17) under this initial configuration is $L_1 \leq \frac{H(s^{(0)})}{H(s^*)}$. Therefore, L_1 does not convey any information about the coverage performance of the obtained PGA solution. This issue is addressed as follows (using the notation $[\cdot]$ and $\{\cdot\}$ to represent a conversion from a set to an array and vice versa).

Once the PGA solution (s^{PGA}, t^{PGA}) is obtained using Alg. 2, it yields information on: (i) optimal agent locations (i.e., s^{PGA}), and, (ii) optimal agent team composition (i.e., t^{PGA}). This allows us to update the discretized feasible space F^D into F^{D2} by inserting the agent coordinates found in s^{PGA} such that $F^{D2} \triangleq F^D \cup \{s^{PGA}\}$. Next, we re-evaluate the greedy algorithm considering only the agents in the optimal team and using the modified discretized

Algorithm 2 Projected Gradient Ascent (PGA) Algorithm for solving the problem in (7).

1: **Inputs:** Ω , F , N , ε_s , ε_t ; **Outputs:** s^{PGA} , t^{PGA} .
2: **Initialize:** $s^0 := [S^G]$ (From Alg. 1), $t^0 \in \mathbb{R}^n$.
3: **for** $k = 0, 1, 2, \dots$ **do**
4: **Compute:** At $(s, t) = (s^{(k)}, t^{(k)})$ the gradients:
 $\frac{\partial H(s, t)}{\partial s_i}$ and $\frac{\partial H(s, t)}{\partial t_i}$, $\forall i \in \{1, \dots, N\}$. ▷ Using (18), (19).
5: **Update:** $(s^{(k)}, t^{(k)})$, by, $\forall i \in \{1, \dots, N\}$,
 $\hat{s}_i^{(k+1)} = s_i^{(k)} + \eta_s^{(k)} \frac{\partial H(s, t)}{\partial s_i}$; $\hat{t}_i^{(k+1)} = t_i^{(k)} + \eta_t^{(k)} \frac{\partial H(s, t)}{\partial t_i}$; (20)
6: **Project:** to get $(s^{(k+1)}, t^{(k+1)})$, $\forall i \in \{1, \dots, N\}$,
 $s_i^{(k+1)} = \Pi_F(\hat{s}_i^{(k+1)})$; $t_i^{(k+1)} = \Pi_{[0, 1]}(\hat{t}_i^{(k+1)})$;
7: **if** $\|s^{(k+1)} - s^{(k)}\| \leq \varepsilon_s$ & $\|t^{(k+1)} - t^{(k)}\| \leq \varepsilon_t$ **then**
8: $(s^{PGA}, t^{PGA}) := (s^{(k+1)}, t^{(k+1)})$; **Return**;
9: **end if** ▷ ε_s and ε_t are positive tolerances.
10: **end for**

feasible space F^{D2} . Now, if the corresponding greedy solution is S^{G2} and the overall performance bound is L_2 (obtained from (17)), following (12) we can write $L_2 \leq \frac{H(S^{G2})}{H(s^*)}$. This relationship together with $H(\{s^{PGA}\})$ can then be used to impose a lower bound to the ratio $\frac{H(\{s^{PGA}\})}{H(s^*)}$ as follows:

$$L' \triangleq L_2 \cdot \frac{H(\{s^{PGA}\})}{H(S^{G2})} \leq \frac{H(\{s^{PGA}\})}{H(s^*)}. \quad (21)$$

Hence L' is used as a performance bound on the final coverage level achieved by the chosen optimal team of agents.

Characterization of optimal t_i values given by PGA. We consider two agents i and j to be *neighbors* if their sensing regions overlap (i.e., $V(s_i) \cap V(s_j) \neq \emptyset$). The set of neighbors of agent i is denoted by $B_i = \{j : j \neq i, V(s_i) \cap V(s_j) \neq \emptyset\}$ and the *closed neighborhood* of agent i is defined as $\bar{B}_i = B_i \cup \{i\}$. Using these neighborhood concepts, we define the following state variable compositions to go along with (s_i, t_i) : (i) The *neighbor state* variables: $(\bar{s}_i^c, \bar{t}_i^c)$, where $\bar{s}_i^c = [\{s_j : j \in B_i\}]$ and $\bar{t}_i^c = [\{t_j : j \in B_i\}]$; (ii) The *neighborhood state* variables: $(\bar{s}_i, \bar{t}_i)$, where $\bar{s}_i = [\{s_j : j \in \bar{B}_i\}]$ and $\bar{t}_i = [\{t_j : j \in \bar{B}_i\}]$; (iii) The *complementary state* variables: (s_i^c, t_i^c) , where $s_i^c = [\{s_j : \forall j \neq i\}]$ and $t_i^c = [\{t_j : \forall j \neq i\}]$. Using this notation, we now establish the following lemma.

Lemma 1. The objective function $H(s, t)$ in (7) can be decomposed as,

$$H(s, t) = t_i H_i(\bar{s}_i, \bar{t}_i^c) + H_i^c(s_i^c, t_i^c) \quad (22)$$

where $H_i(\bar{s}_i, \bar{t}_i^c) = \int_{V(s_i)} R(x) \Phi_i(x) p_i(x, s_i) dx - \beta \gamma_i$, and $H_i^c(s_i^c, t_i^c) = \int_{\Omega} R(x) [1 - \prod_{i \neq j} (1 - \bar{p}_i(x, s_i, t_i))] dx - \beta \sum_{i \neq j} \gamma_i t_i$.

Proof. $H(s, t)$ as given in (7) can be expanded as $H(s, t) = \int_{\Omega} R(x) (1 - (1 - \bar{p}_i(x, s_i, t_i)) \prod_{i \neq j} (1 - \bar{p}_i(x, s_i, t_i))) dx - \beta \gamma_i t_i - \beta \sum_{i \neq j} \gamma_i t_i$. Now, using the following relationships (directly obtained from (5), (2), and from the definition of the neighbor set B_i): (i) $\bar{p}_i(x, s_i, t_i) = t_i p_i(x, s_i)$, $\forall x, s_i \in \Omega$, $\forall t_i \in [0, 1]$, (ii) $p_i(x, s_i) = 0$, $\forall s_i \in \Omega$, $x \notin V(s_i)$, (iii) $p_i(x, s_i)(1 - p_j(x, s_j)) = p_i(x, s_i)$, $\forall x \in \Omega$, $\forall j \notin B_i$, we can write, for all $x, s_i, s_j \in \Omega$ and $t_i, t_j \in [0, 1]$, $\bar{p}_i(x, s_i, t_i) \prod_{i \neq j} (1 - \bar{p}_i(x, s_i, t_i)) = t_i p_i(x, s_i) \prod_{i \in B_i} (1 - \bar{p}_i(x, s_i, t_i))$. Using these two main results, the relationship in (22) can be obtained. ■

We can now establish the following theorem which characterizes the nature of t_i^* , the t_i values given by the PGA Alg. 2.

Table 1
Different classes of agents.

Class	Index	Sensing Para.		Case I $w_{2i} = 1$, Cost (γ_i)	Case II $\gamma_i = 30\,175$, Weight (w_{2i})
		Range (δ_i)	Decay (λ_i)		
1	1 ~ 5	200	0.012	30 175	1.000
2	6 ~ 10	100	0.008	18 772	1.607

Theorem 2. For any agent i , the values obtained from the PGA algorithm satisfy

$$t_i^* = \begin{cases} 0 & \text{when } H_i(\bar{s}_i^*, \bar{t}_i^{c*}) < 0, \\ 1 & \text{when } H_i(\bar{s}_i^*, \bar{t}_i^{c*}) > 0. \end{cases} \quad (23)$$

Moreover, when $H_i(\bar{s}_i^*, \bar{t}_i^{c*}) = 0$, the optimal objective function value $H(\mathbf{s}^*, \mathbf{t}^*)$ is invariant to t_i^* .

Proof. Using the decomposition shown in Lemma 1, we get $\frac{\partial H(\mathbf{s}, \mathbf{t})}{\partial t_i} = H_i(\bar{s}_i, \bar{t}_i^c)$, where $H_i(\bar{s}_i, \bar{t}_i^c)$ is independent of t_i . Therefore, when $H_i(\bar{s}_i, \bar{t}_i^c) \neq 0$, it is clear that the PGA cannot terminate the t_i update process in (Eq. (20)) until t_i hits a constraint boundary given by $t_i \in [0, 1]$. The update direction depends on the sign of $H_i(\bar{s}_i, \bar{t}_i^c)$ and update process in (Eq. (20)) will become stationary when t_i satisfies (23). To prove the second statement, consider the case where $H_i(\bar{s}_i^*, \bar{t}_i^{c*}) = 0$ with $t_i^* \in (0, 1)$. Since $H_i(\bar{s}_i, \bar{t}_i^c)$ is independent of t_i , if t_i^* is perturbed to a value $t_i = t_i^* + \Delta \in [0, 1]$, the optimality condition $H_i(\bar{s}_i^*, \bar{t}_i^{c*}) = 0$ still holds true. Further, using this relationship with Lemma 1, we can see that $H(\mathbf{s}, \mathbf{t})$ is insensitive to a perturbation $t_i^* + \Delta \in [0, 1]$ when at $(\mathbf{s}, \mathbf{t}) = (\mathbf{s}^*, \mathbf{t}^*)$. This means that if the PGA converges to a value $t_i = t_i^* \in (0, 1)$, perturbing t_i towards either 0 or 1 will not affect the objective function value. ■

In conclusion, the proposed PGA ensures that the resulting optimal t_i values are either 0 or 1. Hence, despite the relaxation of the binary variable t_i to $t_i \in [0, 1]$, it provides a solution to the mixed integer non-linear programming problem version of (7), where, for all i , t_i is constrained to $t_i \in \{0, 1\}$. We conclude this section by observing that Lemma 1 makes it clear that in order for an agent to compute the gradients, it only needs the neighborhood state information $(\bar{s}_i, \bar{t}_i)$. Therefore, in executing the PGA, agents have the capability to perform all required computations (and subsequent actuations) in a distributed manner.

5. Numerical results

In this section, we provide several numerical results obtained from the proposed PGA (Alg. 2) initialized with the solution provided by the greedy Alg. 1 discussed in Section 3. The PGA method is evaluated under four different mission space configurations named: (i) General, (ii) Room, (iii) Maze, and, (iv) Narrow, as shown in Figs. 4a, 4b, 4c and 4d, respectively. The mission space is a square of size 600×600 units with an event density function $R(x)$ assumed to be uniform (i.e., $R(x) = 1, \forall x \in F$). All simulations are initialized with ten agents (i.e., $N = 10$) and each agent's nominal sensing capacity is selected as $p_{i0} = 1$. For the use of the greedy algorithm, the ground set F^D is constructed by uniformly placing 100 points in the mission space. All reported simulation results and execution times have been obtained by executing the algorithms on a standard desktop computer with 8.0 GB RAM and a 3.61 GHz AMD eight-core processor. For convenience, we define the cost component of the overall objective function $H(\mathbf{s}, \mathbf{t})$ as $C(t) = \beta \sum_{i=1}^N \gamma_i t_i$. Therefore, $H(\mathbf{s}, \mathbf{t}) = H(\mathbf{s}) - C(t)$ where $H(\mathbf{s})$ represents the coverage component of $H(\mathbf{s}, \mathbf{t})$.

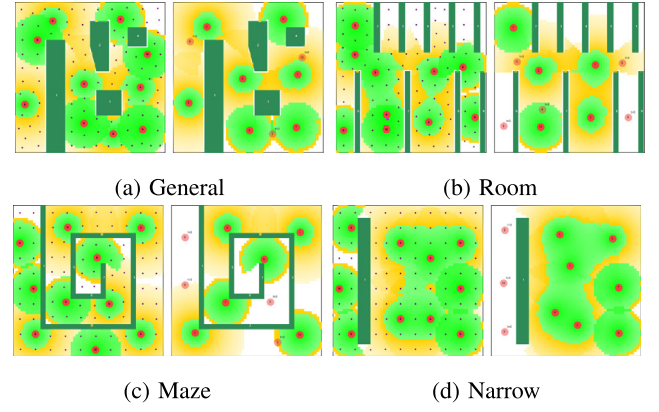


Fig. 4. Comparison of initial greedy solution (left) and projected gradient ascent (PGA) algorithm solution (right) under different mission spaces for the heterogeneous agent case.

The heterogeneous agent case (Case I). In this case, agents differ from each other in terms of both sensing parameters (i.e., sensing range δ_i and sensing decay λ_i) and cost parameters (i.e., agent cost γ_i). To create such a heterogeneous agent configuration, we first assume that the initially available 10 agents belong to two classes (5 agents per each class) as given in Table 1. Then, we set the agent cost weights to $w_{2i} = 1 \forall i$. Based on (2), under each adopted agent class, sensing parameters δ_i and λ_i will determine the agent cost γ_i values as shown in Table 1 under the “Case I” column. The normalization weight used is $w_1 = 0.58$. To make the problem meaningful, the agent classes have been chosen so that they have complementary sensing properties.

The results obtained from the PGA algorithm are summarized in Table 2 and the corresponding optimal agent team deployments are shown in Fig. 4, both at the initial greedy step and at the final PGA solution. We can see a significant improvement achieved in $H(\mathbf{s}, \mathbf{t})$ by the PGA steps compared to the initial greedy solution. It is noteworthy that the PGA algorithm has chosen agents from both classes to form the optimal agent team. Also note that, with the help of the initial greedy step, the PGA method has been capable of placing agents in appropriate mission space regions well suited for their specific sensing properties (see agent 6 in Fig. 4a).

The coverage performance bounds L' (defined in (21)) achieved by the optimal agent teams are shown in Table 4. From those results, we can conclude that, on average, the optimal agent team provides more than 75% of the attainable maximum coverage level (slightly less than the average bound observed for the homogeneous agent case).

Sensing-wise heterogeneous agent case (Case II). Our purpose here is to highlight the importance of having different agent costs γ_i when the sensing parameters of the agents are different. We also stress the importance of using the sensing capability (i.e., κ_i) dependent agent costs as proposed in (2). Unlike the previously discussed heterogeneous agent case, here we use a fixed agent cost $\gamma_i = 30\,175$ across all agent classes. To achieve this under (2), we manipulate the agent cost weight w_{2i} parameters in each agent class, as given in Table 1 column “Case II”. As a result of this manipulation, despite the differences in sensing parameters over different agents, the agent costs γ_i across all agents become identical. The normalization weight used is $w_1 = 0.58$.

Since all the other problem settings are identical to the previously discussed heterogeneous agent Case I, the initial greedy step of the PGA algorithm will yield the same agent deployment. However, the associated total agent cost $C(t)$ is different due to

Table 2

Results of the proposed PGA for the heterogeneous agent case.

Mission space	Initial greedy solution				Final PGA solution				Fig.
	$N = 10$	$H(s)$	$C(t)$	$H(s, t)$	Agent team	$H(s)$	$C(t)$	$H(s, t)$	
General	10	152,272	177,140	−22,868	{1, 2, 4, 5}, {6, 7, 8}	124,194	128,174	−3980	4a
Room	10	142,859	177,140	−34,281	{1, 2, 3}, {7, 10}	94,417	92,781	1635	4b
Maze	10	146,175	177,140	−30,965	{1, 2, 4}, {6, 7, 10}	96,889	106,355	−9465	4c
Narrow	10	179,478	177,140	2337	{1, 2, 3, 4, 5}, {6, 7}	145,963	136,420	9543	4d

Table 3

Results of the proposed PGA for the sensing-wise heterogeneous agent case.

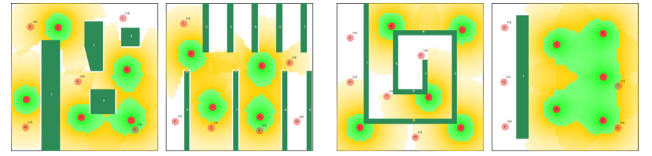
Mission space	Initial greedy solution				Final PGA solution				Fig.
	N	$H(s)$	$C(t)$	$H(s, t)$	Agent team	$H(s)$	$C(t)$	$H(s, t)$	
General	10	156,142	177,140	−20,997	{1, 2, 3, 4, 5}, {}	97,398	88,671	8726	5a
Room	10	145,848	177,140	−31,292	{1, 2, 3, 5}, {}	79,771	70,972	8798	5b
Maze	10	146,175	177,140	−30,975	{1, 2, 3, 4, 5}, {}	83,261	88,671	−5410	5c
Narrow	10	179,478	177,140	2337	{1, 2, 3, 4, 5}, {}	120,374	88,671	31,703	5d

Table 4Performance bound guarantees (i.e., L' in (21)) on the final coverage level achieved by the optimal agent team for the heterogeneous agent case.

Mission space	Agent team	$H(S^{G2})$	L_2	L'
General	{2, 3, 4, 5}, {6, 9, 10}	117,923	0.703	0.740
Room	{1, 2, 3}, {7, 10}	86,534	0.853	0.931
Maze	{1, 2, 4}, {6, 7, 10}	91,203	0.703	0.747
Narrow	{1, 2, 3, 4, 5}, {6, 7}	144,852	0.651	0.656

Table 5Performance bound guarantees (i.e., L' in (21)) on the final coverage level achieved by the optimal agent team for the sensing-wise heterogeneous agent case.

Mission space	Agent team	$H(S^{G2})$	L_2	L'
General	{1, 2, 3, 4, 5}, {}	95,633	0.729	0.742
Room	{1, 2, 3, 5}, {}	73,864	0.813	0.878
Maze	{1, 2, 3, 4, 5}, {}	82,957	0.703	0.706
Narrow	{1, 2, 3, 4, 5}, {}	117,231	0.651	0.668



(a) General (b) Room (c) Maze (d) Narrow

Fig. 5. The obtained final PGA solution under different mission spaces for the sensing-wise heterogeneous agent case.

the modification of agent cost parameters w_{2i} and γ_i compared to that of the heterogeneous agent case. The numerical results obtained are summarized in Table 3 and the optimal agent team deployments are shown in Fig. 5. The coverage performance bounds L' (defined in (21)) achieved by the optimal agent team are tabulated in Table 5. As expected, when identical agent costs are used despite their differences in sensing capabilities, the resulting PGA solution gives preference to agents with higher sensing capabilities. Thus, the optimal agent team is inherently biased towards Class 1 agents (see Table 3 and notice $\kappa_1 > \kappa_2$ due to the δ_i, λ_i values $i = 1, 2$). Clearly, in real-world applications one expects more capable sensors to have higher costs.

Comparison with a commercial optimization solver. In comparing the solutions given by the proposed PGA method to those of a commercially available optimization problem solver, there are two constraining factors to consider: (i) The coverage component of the objective function in (7) is non-convex, non-linear, and discontinuous. As a result, even though the original version of (7) is a mixed-integer non-linear program (MINLP) (where $t_i \in \{0, 1\}$, $\forall i$), we were constrained to using a generic non-linear program (NLP) solver. Therefore, in order to find the optimal binary decision variables (i.e., t), we applied the NLP solver exhaustively over all possible integer values (we refer to this as the “brute force” method). (ii) When obstacles are present in the mission space, the feasible space for each agent becomes non-convex (in our case, this complicates the objective function as well). Since representing such constraints and feeding them to a generic optimization problem solver is difficult, we confine our study to an obstacle-less (blank) mission space.

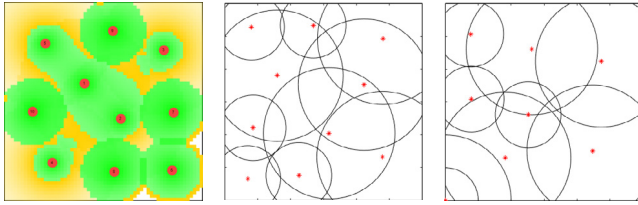
The NLP solver used is the *interior point method* implemented under the *fmincon* command in MATLAB®. The available agents and their sensing capabilities are given in Table 1. In the brute force approach, each iteration considers a specific agent team and computes the optimal coverage solution. Two brute force methods (BF₁ and BF₂) were used depending on the agent initialization in order to highlight the effect of such initialization. Specifically, in BF₁, agents are initialized randomly and in BF₂, agents are initialized in a corner of the mission space such that the l th agent ($\forall l$) is placed at $s_l = (5 + 5l, 5 + 5l)$. Note that when the normalization weight is $w_1 = 1$ (see (8), (7)), the PGA method basically solves the optimal coverage problem. This enables a direct comparison of the performance of the PGA method (when $w_1 = 1$) with that of single iterations of BF₁ and BF₂. This comparison is shown in Fig. 6 and it confirms that the proposed PGA method: (i) Delivers better coverage levels, and, (ii) Shows extremely low execution times compared to BF₁ or BF₂. Another conclusion is that the random initialization has helped the BF₁ method to achieve better coverage and execution times compared to that of BF₂.

Under the information in Table 1, there are 35 possible agent team configurations. Therefore, 35 brute force iterations were required to determine the optimal agent configuration. As the next step, the agent cost related parameters β and γ_i were computed using the prespecified weights w_1 and w_{2i} . Then, the best agent team composition, which maximizes the overall objective $H(s, t)$, is identified from simply searching through the previously generated results. A comparison of the obtained results in terms of the coverage $H(s)$ and the overall objective $H(s, t)$ when the weight w_1 is varied is shown in Fig. 7. The average value of the execution times observed in each method is given in Table 6.

Our main conclusions from this comparison are: (i) The PGA method delivers better coverage levels $H(s)$ across all w_1 values used, and, (ii) As w_1 increases, the PGA method performs better than brute force methods in terms of $H(s, t)$, and, most importantly, (iii) The average execution time required for the PGA

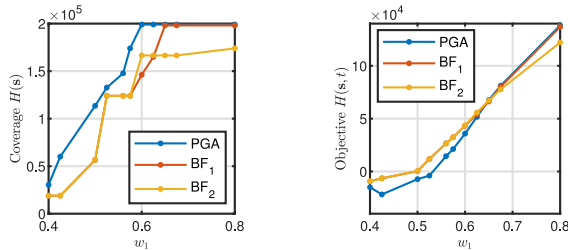
Table 6
Observed average execution times.

Method	PGA	BF ₁	BF ₂
Average execution time (s)	4.56	4328.13	8845.83



(a) PGA ($w_1 = 1$):
 $H(s) = 199037$
Ex.T. = 2.699s
(b) BF₁ (1 Iter.):
 $H(s) = 198083$
Ex.T. = 246.523s
(c) BF₂ (1 Iter.):
 $H(s) = 164345$
Ex.T. = 728.374

Fig. 6. Optimal agent configurations, coverage levels, and execution times obtained for the multi-agent coverage problem (see (3)) with 10 heterogeneous agents (see Table 1) in a blank mission space using (a) PGA algorithm, (b) Brute force method 1 (BF₁), and, (c) Brute force method 2 (BF₂).



(a) Coverage performance: $H(s)$ (b) Overall performance: $H(s, t)$

Fig. 7. Comparison of coverage performance $H(s)$ and overall performance $H(s, t)$ for different normalization weights w_1 in (8).

method is extremely low compared to brute force approaches (by a factor of 10^{-3}). Finally, we also emphasize the scalability that the PGA method offers due to its distributed nature.

6. Conclusions

We address the multi-agent coverage problem where the number of agents to be used is flexible and the available agents are both heterogeneous and have an associated cost value. We have addressed this optimal agent team composition problem by constructing an objective function combining the overall agent team cost with the coverage level delivered by the agent team. This problem is then solved using a projected gradient ascent (PGA) algorithm initialized through a greedy algorithm and shown to recover the integer-valued variables that were originally relaxed. Further, based on submodularity theory, we have derived tighter performance bounds showing that the PGA algorithm can often lead to near-global-optimal solutions. The effectiveness and computational efficiency of the PGA algorithm in diverse mission spaces and heterogeneous multi-agent scenarios has been validated.

References

- Balmann, A. (2000). Modeling land use with multi-agent systems: Perspectives for the analysis of agricultural policies. In *Intl. institute of fisheries economics and trade proceedings* (p. 11).
Bertsekas, D. P. (2016). *Nonlinear programming*. Athena Scientific.
Breitenmoser, A., Schwager, M., Metzger, J.-C., Siegwart, R., & Rus, D. (2010). Voronoi coverage of non-convex environments with a group of networked robots. In *2010 IEEE Intl. Conf. on robotics and automation* (pp. 4982–4989).

- Caicedo-Nunez, C. H., & Zefran, M. (2008). A coverage algorithm for a class of non-convex regions. In *47th IEEE Conf. on decision and control* (pp. 4244–4249).
Cassandras, C. G., & Li, W. (2005). Sensor networks and cooperative control. In *44th IEEE Conf. on decision and control*, Vol. 2005 (pp. 4237–4238).
Castanedo, F., García, J., Patricio, M. A., & Molina, J. M. (2010). Data fusion to improve trajectory tracking in a cooperative surveillance multi-agent architecture. *Information Fusion*, 11(3), 243–255.
Conforti, M., & Cornuéjols, G. (1984). Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the Rado-Edmonds theorem. *Discrete Applied Mathematics*, 7(3), 251–274.
Cortes, J., Martinez, S., Karatas, T., & Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2), 243–255.
Davis, L. (1996). *Handbook of genetic algorithms*. Boston: London International Thomson Computer Press.
Fisher, M. L., Nemhauser, G. L., & Wolsey, L. A. (1978). An analysis of approximations for maximizing submodular set functions—II. *Polyhedral Combinatorics: Dedicated to the Memory of D.R. Fulkerson*, 73–87.
Gusrialdi, A., Hirche, S., Hatanaka, T., & Fujita, M. (2008). Voronoi based coverage control with anisotropic sensors. In *American control conference* (pp. 736–741).
Gusrialdi, A., & Zeng, L. (2011). Distributed deployment algorithms for robotic visual sensor networks in non-convex environment. In *2011 Intl. Conf. on networking, sensing and control* (pp. 445–450).
Liu, Y., Chong, E. K. P., & Pezeschi, A. (2018). Improved bounds for the greedy strategy in optimization problems with curvature. *Journal of Combinatorial Optimization*, 37(4), 1126–1149.
Luo, C., Espinosa, A. P., Pranantha, D., & De Gloria, A. (2011). Multi-robot search and rescue team. In *IEEE Intl. Symp. on safety, security, and rescue robotics* (pp. 296–301).
Meguerdichian, S., Koushanfar, F., Potkonjak, M., & Srivastava, M. B. (2001). Coverage problems in wireless ad-hoc sensor networks. In *IEEE INFOCOM* (pp. 1380–1387).
Nemhauser, G. L., Wolsey, L. A., & Fisher, M. (1978). An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1), 265–294.
Schwager, M., Bullo, F., Skelly, D., & Rus, D. (2008). A ladybug exploration strategy for distributed adaptive coverage control. In *2008 IEEE Intl. Conf. on robotics and automation* (pp. 2346–2353).
Sun, X., Cassandras, C. G., & Gokbayrak, K. (2014). Escaping local optima in a class of multi-agent distributed optimization problems: A boosting function approach. In *53rd IEEE Conf. on decision and control* (pp. 3701–3706).
Sun, X., Cassandras, C. G., & Meng, X. (2019). Exploiting submodularity to quantify near-optimality in multi-agent coverage problems. *Automatica*, 100, 349–359.
Sun, C., Welikala, S., & Cassandras, C. G. (2019). Optimal composition of heterogeneous multi-agent teams for coverage problems with performance bound guarantees. [Online]. Available: <http://arxiv.org/abs/2003.05587>.
Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, 58(1), 267–288.
Wang, Z., Moran, B., Wang, X., & Pan, Q. (2016). Approximation for maximizing monotone non-decreasing set functions with a greedy method. *Journal of Combinatorial Optimization*, 31(1), 29–43.
Welikala, S., & Cassandras, C. G. (2020). Distributed non-convex optimization of multi-agent systems using boosting functions to escape local optima. In (Accepted) *American control conference*.
Zheng, Y., Ma, J., & Wang, L. (2018). Consensus of hybrid multi-agent systems. *IEEE Transactions on Neural Networks and Learning Systems*, 29(4), 1359–1365.
Zhong, M., & Cassandras, C. G. (2011). Distributed coverage control and data collection with mobile sensor networks. *IEEE Transactions on Automatic Control*, 56(10), 2445–2455.



Chuangchuang Sun is currently a postdoctoral associate in the department of aeronautics and astronautics at MIT. He received his Ph.D. in August 2018 from the Ohio State University and a B.S. degree from the Beijing University of Aeronautics and Astronautics, China in 2013 both in Aerospace Engineering. His research interest focus on control, optimization, reinforcement learning and applications in robotics and space systems.



Shirantha Welikala received the B.Sc. Eng. degree in electrical and electronic engineering from the University of Peradeniya, Sri Lanka, in 2015. From 2015 to 2017, he was with the Department of Electrical and Electronic Engineering, University of Peradeniya, Sri Lanka, where he worked first as a Temporary Instructor and subsequently as a Research Assistant. He is currently pursuing the Ph.D. degree in Systems Engineering with Boston University, Brookline, MA, USA. His main research interests include optimization and control of multiagent systems, robotics, signal processing, and smart grid applications.



Christos G. Cassandras is Distinguished Professor of Engineering at Boston University. He is Head of the Division of Systems Engineering, Professor of Electrical and Computer Engineering, and cofounder of Boston University's Center for Information and Systems Engineering (CISE). He received degrees from Yale University, Stanford University and Harvard University. In 1982–84 he was with ITP Boston, Inc. where he worked on the design of automated manufacturing systems. In 1984–1996 he was a faculty member at the Department of Electrical and Computer Engineering

University of Massachusetts/ Amherst. He specializes in the areas of discrete event and hybrid systems, cooperative control, stochastic optimization, and computer simulations with applications to computer and sensor networks,

manufacturing systems, and transportation systems. He has published over 400 refereed papers in these areas, and six books. He has guest edited several technical journal issues and currently serves on several journal Editorial Boards, including Editor of Automatica. In addition to his academic activities, he has worked extensively with industrial organizations on various systems integration projects and the development of decision support software. He has most recently collaborated with The MathWorks, Inc. in the development of the discrete event and hybrid system simulator SimEvents.

Dr. Cassandras was Editor in Chief of the IEEE Transactions on Automatic Control from 1998 through 2009 and has also served as Editor for Technical Notes and Correspondence and Associate Editor. He was the 2012 President of the IEEE Control Systems Society (CSS). He has also served as Vice President for Publications and on the Board of Governors of the CSS, as well as on several IEEE committees and has chaired several conferences. He has been a plenary/keynote speaker at numerous international conferences including the 2017 IFAC World Congress the American Control Conference in 2001 and the IEEE Conference on Decision and Control in 2002 and 2016s and has also been an IEEE Distinguished Lecturer.

He is the recipient of several awards, including the 2011 IEEE Control Systems Technology Awards the Distinguished Member Award of the IEEE Control Systems Society (2006)s the 1999 Harold Chestnut Prize (IFAC Best Control Engineering Textbook) for Discrete Event Systems: Modeling and Performance Analysis, a 2011 prize and a 2014 prize for the IBM/IEEE Smarter Planet Challenge competition, the 2014 Engineering Distinguished Scholar Award at Boston University, several honorary professorships, a 1991 Lilly Fellowship and a 2012 Kern Fellowship. He is a member of Phi Beta Kappa and Tau Beta Pi. He is also a Fellow of the IEEE and a Fellow of the IFAC.