Automatica 100 (2019) 349–359

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Exploiting submodularity to quantify near-optimality in multi-agent coverage problems[☆]

Xinmiao Sun[a,b,*], Christos G. Cassandras[a], Xiangyu Meng[a]

[a] Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA 02446, USA
[b] School of Automation and Electrical Engineering, University of Science and Technology Beijing, 100083, China

ARTICLE INFO

Article history:
Received 23 August 2017
Received in revised form 11 July 2018
Accepted 31 October 2018

Keywords:
Optimal coverage
Multi-agent systems
Submodular set function
Greedy algorithm
Gradient method

ABSTRACT

We consider optimal coverage problems for a multi-agent network aiming to maximize a joint event detection probability in an environment with obstacles. The objective function of this problem is non-concave and no global optimum is guaranteed by gradient-based algorithms developed to date. In order to obtain a solution provably close to the global optimum, the selection of initial conditions is crucial. We first formulate the initial agent location generation as an additional optimization problem where the objective function is monotone submodular, a class of functions for which the performance obtained through a greedy algorithm solution is guaranteed to be within a provable bound relative to the optimal performance. We then derive two tighter bounds by exploiting the curvature information (total curvature and elemental curvature) of the objective function. We further show that the tightness of these lower bounds is complementary with respect to the sensing capabilities of the agents. The greedy algorithm solution can be subsequently used as an initial point of a gradient-based algorithm for the original optimal coverage problem. Simulation results are included to verify that this approach leads to significantly better performance relative to previously used algorithms.

1. Introduction

Multi-agent systems consist of a team of agents, e.g., vehicles, robots, or sensor nodes, that cooperatively perform one or more tasks in a mission space which may contain uncertainties in the form of obstacles or random event occurrences. Examples of such tasks include environmental monitoring, surveillance, or animal population studies among many. Optimization problems formulated in the context of multi-agent systems, more often than not, involve non-convex objective functions resulting in potential local optima, while global optimality cannot be easily guaranteed.

One of the fundamental problems in multi-agent systems is the optimal coverage problem where agents are deployed so as to cooperatively maximize the coverage of a given mission space (Breitenmoser, Schwager, Metzger, Siegwart, & Rus, 2010; Caicedo-Nuez & Zefran, 2008; Caicedo-Nunez & Zefran, 2008; Cassandras & Sensor, 2005; Meguerdichian, Koushanfar, Potkonjak, & Srivastava, 2001) where "coverage" is measured in a variety of ways, often through a joint detection probability of random events cooperatively detected by the agents. The problem can be solved by either on-line or off-line methods. Some widely used on-line methods, such as distributed gradient-based algorithms (Cassandras & Sensor, 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; Kantaros, Thanou, & Tzes, 2015; Marier, Rabbath, & Léchevin, 2012), typically result in locally optimal solutions, hence possibly poor performance. To escape such local optima, a "boosting function" approach is proposed in Sun, Cassandras, and Gokbayrak (2014) whose performance can be ensured to be no less than that of these local optima. Alternatively, a "ladybug exploration" strategy is applied to an adaptive controller in Schwager, Bullo, Skelly, and Rus (2008), which aims at balancing coverage and exploration. However, these on-line approaches cannot quantify the gap between the local optima they attain and the global optimum. Off-line algorithms, such as simulated annealing (Bertsimas & Tsitsiklis, 1993; Van Laarhoven & Aarts, 1987), can, under certain conditions, converge to a global optimal solution in probability. However, they are limited by a high computational load and slow convergence rate.

[☆] This work was supported in part by National Science Foundation, USA under grants ECCS-1509084, IIP-1430145, and CNS-1645681, by Air Force Office of Scientific Research, USA under grant FA9550-12-1-0113, by Advanced Research Projects Agency - Energy's, USA NEXTCAR program under grant DE-AR0000796, and by Bosch and the MathWorks, USA. The material in this paper was partially presented at the 56th IEEE Conference on Decision and Control, December 12–15, 2017, Melbourne, Australia. This paper was recommended for publication in revised form by Associate Editor Giancarlo Ferrari-Trecate under the direction of Editor Ian R. Petersen.

[*] Corresponding author.
E-mail addresses: sxm2233hgt@126.com, xmsun@bu.edu (X. Sun), cgc@bu.edu (C.G. Cassandras), xymeng@bu.edu (X. Meng).

https://doi.org/10.1016/j.automatica.2018.11.020
0005-1098/© 2018 Elsevier Ltd. All rights reserved.

Related to the optimal coverage problem is the "maximum coverage" problem (Berman & Krass, 2002; Khuller, Moss, & Naor, 1999), where a collection of discrete sets is given (the sets may have some elements in common and the number of elements is finite) and at most N of these sets are selected so that their union has maximal size (cardinality). The objective function in the maximum coverage problem is *submodular*, a special class of set functions with attractive properties one can exploit. In particular, a well known result in the submodularity theory (Nemhauser, Wolsey, & Fisher, 1978) is the existence of a lower bound for the global optimum provided by any feasible solution obtained by the *greedy algorithm*, i.e., an algorithm which iteratively picks the set that covers the maximum number of uncovered elements at each iterative step. Defining, for any integer number N of sets, $L(N) = f/f^\star$ where $f^\star$ is the global optimum and f is a feasible solution obtained by the greedy algorithm, it is shown in Nemhauser et al. (1978) that $L(N) \geq 1 - \frac{1}{e} \simeq 0.63212$. In other words, since $f^\star \leq (1 - \frac{1}{e})^{-1} f$, one can quantify the optimality gap associated with a given solution f.

In our past work (Sun et al., 2014), we studied the optimal coverage problem with agents allowed to be positioned at any feasible point in the mission space (which generally includes several obstacles) and used a distributed gradient-based algorithm to determine optimal agent locations. Depending on initial conditions, a trajectory generated by such gradient-based algorithms may lead to a local optimum. In this paper, we begin with an initial agent location generation by solving an additional optimization problem with the same coverage metric but limiting agents to a finite set of feasible positions. An advantage of this formulation is that it assists us in eliminating obviously bad initial conditions for any gradient-based method. An additional advantage comes from the fact that we can show our coverage objective function to be monotone submodular, therefore, a suboptimal solution obtained by the greedy algorithm can achieve a performance ratio $L(N) \geq 1 - \frac{1}{e}$, where N is the number of agents in the system. The idea of exploiting the submodularity of the objective function in optimization problems has been used in the literature, e.g., in sensor placement (Krause, Leskovec, Guestrin, VanBriesen, & Faloutsos, 2008; Krause, Singh, & Guestrin, 2008) and the maximum coverage problem mentioned above, whereas a total backward curvature of string submodular functions is proposed in Zhang, Chong, Pezeshki, and Moran (2016) and a total curvature c_k for the k-batch greedy algorithm is proposed in Liu, Zhang, Chong, and Pezeshki (2016) in order to derive bounds for related problems.

Our goal in this paper is to derive a tighter lower bound, i.e., to increase the ratio $L(N)$ defined above by further exploiting the structure of our objective function. In particular, we make use of the *total curvature* (Conforti & Cornuéjols, 1984) and the *elemental curvature* (Wang, Moran, Wang, & Pan, 2016) of the objective function and show that these can be explicitly derived and lead to new and tighter lower bounds. Moreover, we show that the tightness of the lower bounds obtained through the total curvature and the elemental curvature respectively is *complementary* with respect to the sensing capabilities of the agents. In other words, when the sensing capabilities are weak, one of the two bounds is tight and when the sensing capabilities are strong, the other bound is tight. Thus, regardless of the sensing properties of our agents, we can always determine a lower bound tighter than $L(N) = 1 - \frac{1}{e}$ and, in some cases very close to 1, implying that the greedy algorithm solution can be guaranteed to be near-globally optimal.

Another contribution of the paper is to add a final step to the optimal coverage process, after obtaining the greedy algorithm solution and evaluating the associated lower bound with respect to the global optimum. Specifically, we relax the set of allowable agent positions in the mission space from the imposed discrete set and use the solution of the greedy algorithm as an initial condition for the distributed gradient-based algorithm in Sun et al. (2014). We refer to this as the *Greedy-Gradient Algorithm* (GGA) which is applicable to the original coverage problem.

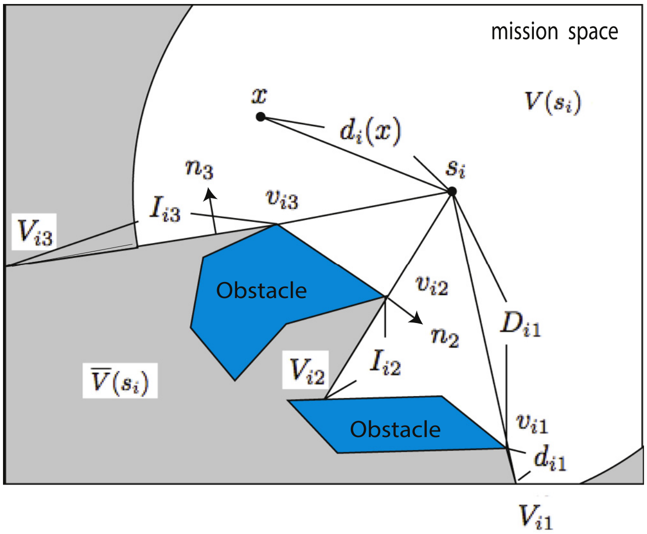


Fig. 1. Mission space with two polygonal obstacles.

The remainder of this paper is organized as follows. The optimal coverage problem is formulated in Section 2. The GGA for the optimal coverage problem is presented in Section 3, where we provide the greedy algorithm and discuss provable optimality bounds of the greedy algorithm solution according to the submodularity theory in Section 3.1, and show how the greedy algorithm and the gradient method can be combined together to solve the optimal coverage problem in Section 3.2. In Section 4, we provide simulation examples to show how the algorithm works and can provide significantly better performance compared to earlier results reported in Sun et al. (2014).

2. Optimal coverage problem formulation

We begin by reviewing the basic coverage problem presented in Caicedo-Nunez and Zefran (2008); Cortes et al. (2004); Zhong and Cassandras (2011). A *mission space* $\Omega \subset \mathbb{R}^2$ is modeled as a non-self-intersecting polygon, i.e., a polygon such that any two non-consecutive edges do not intersect. Associated with Ω, we define a function $R(x) : \Omega \to \mathbb{R}$ to characterize the probability of event occurrences at the location $x \in \Omega$. It is referred to as *event density* and satisfies $0 \leq R(x) \leq 1$ for all $x \in \Omega$ and $\int_\Omega R(x)dx < \infty$. The mission space may contain obstacles modeled as m non-self-intersecting polygons denoted by $M_j, j = 1, \dots, m$, which block the movement as well as the sensing range of an agent. The interior of M_j is denoted by $\mathring{M}_j$ and the overall *feasible space* is $F = \Omega \setminus (\mathring{M}_1 \cup \dots \cup \mathring{M}_m)$, i.e., the space Ω excluding all interior points of the obstacles. We also assume that $R(x) = 0$ for $x \notin F$. There are N agents in the mission space and their positions are defined by a vector $\mathbf{s} = (s_1, \dots, s_N)$ with $s_i \in F, i = 1, \dots, N$. We assume that $s_i \neq s_j$ for any two distinct agents i and j. Fig. 1 shows a mission space with two obstacles and an agent located at s_i.

In the coverage problem, agents are viewed as sensor nodes. We assume that each node has a bounded sensing range captured by the *sensing radius* δ_i. Thus, the sensing region of node i is $\Omega_i = \{x : d_i(x) \leq \delta_i\}$, where $d_i(x) = \|x - s_i\|$. The presence of obstacles inhibits the sensing ability of a node, which motivates the definition of a *visibility set* $V(s_i) \subset F$. A point $x \in F$ is *visible*

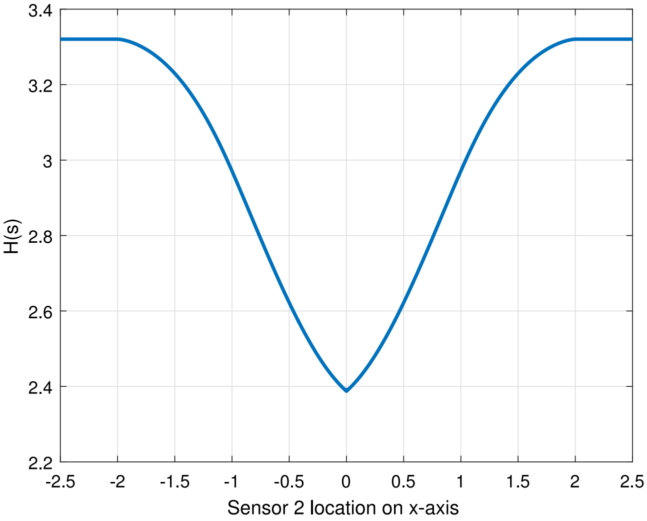


Fig. 2. Example of objective function $H(\mathbf{s})$.

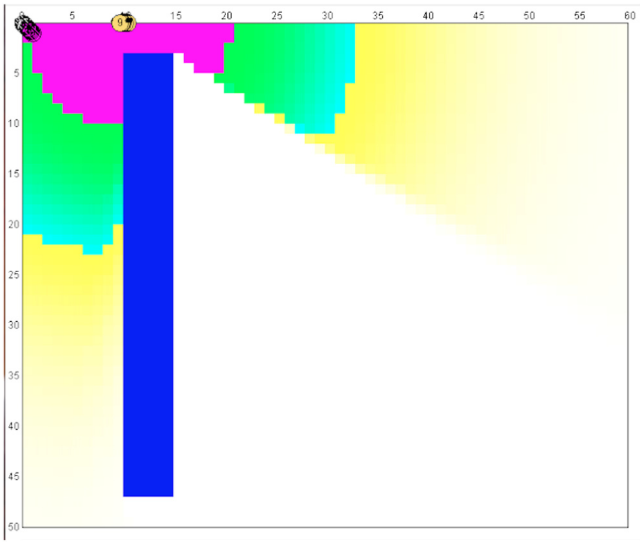


Fig. 3. Example of obviously poor coverage performance.

from $s_i \in F$ if the line segment defined by x and s_i is contained in F, i.e., $\eta x + (1-\eta)s_i \in F$ for all $\eta \in [0, 1]$, and x is within the sensing range of s_i, i.e. $x \in \Omega_i$. Then, $V(s_i) = \Omega_i \cap \{x : \eta x + (1-\eta)s_i \in F \text{ for all } \eta \in [0, 1]\}$ is a set of points in F which are visible from s_i. We also define $\bar{V}(s_i) = F \setminus V(s_i)$ to be the *invisibility set* from s_i, e.g., the gray area in Fig. 1. A sensing model for node i is given by the probability that sensor i detects an event occurrence at $x \in V(s_i)$, denoted by $p_i(x, s_i)$. We assume that $p_i(x, s_i)$ can be expressed as a function of $d_i(x) = \|x - s_i\|$ and is monotonically decreasing and differentiable. An example of such a function is

$$p_i(x, s_i) = \exp(-\lambda_i \|x - s_i\|), \tag{1}$$

where λ_i is a *sensing decay* factor. For points that are invisible to node i, the detection probability is zero. Thus, the overall *sensing detection probability*, denoted by $\hat{p}_i(x, s_i)$, is defined as

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

which is not a continuous function of s_i. Note that $V(s_i) \subset \Omega_i = \{x : d_i(x) \le \delta_i\}$ is limited by the sensing range δ_i of agent i and that the overall sensing detection probability of agents is determined by the sensing range δ_i as well as sensing decay rate λ_i. Then, the *joint detection probability* that an event at $x \in \Omega$ is detected by the N nodes is given by

$$P(x, \mathbf{s}) = 1 - \prod_{i=1}^{N} [1 - \hat{p}_i(x, s_i)], \tag{3}$$

where we assume that detection probabilities of different sensors are independent. The optimal coverage problem can be expressed as follows:

$$\begin{aligned} \max_{\mathbf{s}} \; & H(\mathbf{s}) = \int_{\Omega} R(x)P(x, \mathbf{s})dx \\ \text{s.t.} \; & s_i \in F, \; i = 1, \ldots, N. \end{aligned} \tag{4}$$

We emphasize again that $H(\mathbf{s})$ is not convex (concave) even in the simplest possible problem setting as shown in Fig. 2, where agent 1 is fixed at the origin of a two-dimensional space, and agent 2 moves along the x-axis. Furthermore, the feasible set F is not convex when there are obstacles in the mission space. Therefore, the optimal coverage problem (4) is not a convex optimization problem and may have multiple local optimal points. As a direct consequence of the non-convexity, a gradient method may converge to local optima and possibly result in poor coverage performance. This is evident in Fig. 3 when $R(x)$ is uniform, where all 10 agents converge to a local optimum above the obstacle shown (and remain stuck there), yielding obviously poor coverage performance due to a poor choice of an initial location estimate (or simply a fixed and uncontrollable initial agent location). The mission space is colored from dark to light as the joint detection probability (our objective function) decreases: the joint detection probability is ≥ 0.97 for purple areas, ≥ 0.50 for green areas, and near zero for white areas.

The above observations motivate us to revisit the optimal coverage problem and seek solutions that are provably closer to the global optimum even though determining this global optimum is generally infeasible.

3. Greedy-Gradient algorithm

We propose a two-step approach, termed as *Greedy-Gradient Algorithm* (GGA), to solve the optimal coverage problem. As the name suggests, the first step is to generate desirable initial agent locations by using a greedy algorithm (Section 3.1), and the second step is to obtain solutions even closer to the global optimum by a gradient method (Section 3.2).

3.1. Greedy algorithm

In order to apply the greedy algorithm, the continuous mission space F has to be approximated by a discrete set $F^D = \{f_1, \ldots, f_n\}$ with a finite number of feasible positions such that $f_i \in F$. A general discretization method to approximate the continuous space is given as follows. Horizontal and vertical lines with a line distance Δ are uniformly placed in the mission space to form a grid. The intersecting points which happen to lie within obstacles are discarded. For every other intersecting point x in the grid, we evaluate $R(x)$ based on the point value. Then, we generate a random number r uniformly distributed over $[0, 1]$. If $R(x) > r$, then x is added to F^D; otherwise, x is discarded. In this way, the discretization leverages $R(x)$ such that points with high event density are more likely to be included in F^D.

After the discretization process, we have the following optimization problem:

$$\max_{\mathbf{s}} H(\mathbf{s}) = \int_{\Omega} R(x)P(x, \mathbf{s})dx \qquad \text{s.t.} \quad \mathbf{s} \in \mathcal{I} \tag{5}$$

where $\mathcal{I} = \{S \subseteq F^D : |S| \leq N\}$ is a collection of subsets of F^D, $|S|$ denotes the cardinality of set S.

Remark 1. Even though the objective function $H(\mathbf{s})$ in (5) has the same form as the one in (4), it is a set function instead of a function of a $2N$-dimensional vector. Loosely speaking, the gap between the optimal values of problems (4) and (5) decreases monotonically as Δ decreases.

The problem (5) is a combinatorial optimization problem by choosing N agent positions from n feasible positions, where the time complexity is $n!/(N!(n-N)!)$. A naive method to find the global optimum of (5) is the brute-force search, which may not generate quality solutions in a reasonable amount of time when n and N are large. Finding the optimal solution to (5) is in general NP-hard. The greedy algorithm (described in Section 1 and shown in Algorithm 1) is essentially the best-possible approximation algorithm in polynomial time. The time complexity of the greedy algorithm is $O(nN)$. Note that when $n \gg N$, the time complexity is $O(n)$, which is independent of the number of agents. The following greedy algorithm is used to obtain a feasible solution for (5). The basic idea of the greedy algorithm is to add an agent which can maximize the value of the objective function at each iteration.

Algorithm 1 *Greedy Algorithm*

```
Input: Set function $H(\mathbf{s})$
       Cardinality constraint $N$
Output: Set $\mathbf{s}$
Initialization: $\mathbf{s} \leftarrow \emptyset$, $i \leftarrow 0$
while $i \leq N$ do
    $s_i^* = \arg\max_{s_i \in \mathcal{I} \setminus \mathbf{s}} H(\mathbf{s} \cup \{s_i\})$
    $\mathbf{s} \leftarrow \mathbf{s} \cup \{s_i^*\}$
    $i \leftarrow i + 1$
end while
return $\mathbf{s}$
```

The initial agent locations may be further improved by an "enhanced greedy" algorithm at the expense of additional computational cost. Based on the output of Algorithm 1, agent i's location can be re-selected according to

$$s_i^* = \arg \max_{\bar{s}_i \in \mathcal{I} \setminus s \cup \{s_i\}} H\left(s \setminus \{s_i\} \cup \bar{s}_i\right)$$

given the selected initial locations of all other agents. This process can be iterated for all agents, and repeated several times. Note that even though we cannot guarantee improvements by the enhanced greedy algorithm, it will obviously perform no worse by simply adding a few more iterations to the original greedy algorithm.

Fortunately, our objective function $H(\mathbf{s})$ in (5) can be shown (Theorem 1) to be *monotone submodular*. Therefore, we can apply basic results from submodularity theory, which hold for this class of functions, to show that the greedy algorithm produces a guaranteed performance. In the following, we will introduce the basic elements of submodularity theory and show that submodularity enables the greedy algorithm with provable optimality bounds.

Remark 2. It is worth mentioning that sometimes the coverage problem is stated as (5) rather than (4), i.e., the mission space has already been discretized. This is true, for instance, when we have no capability to perfectly position agents in a continuous space, but are only allowed to place them in a set of a priori given discrete "cells". For this case, it has been shown in Sun, Cassandras, and Meng (2017) that the greedy algorithm actually provides tight bounds with respect to the optimization problem (5).

3.1.1. *Monotone submodular coverage metric*

A submodular function is a set function whose value has the diminishing returns property. The formal definition of submodularity is given as follows (Nemhauser et al., 1978).

Definition 1. Given a ground set $Y = \{y_1, \ldots, y_n\}$ and its power set 2^Y, a function $f : 2^Y \to \mathbb{R}$ is called *submodular* if for any $S, T \subseteq Y$,

$$f(S \cup T) + f(S \cap T) \leq f(S) + f(T). \tag{6}$$

If, additionally, $f(S) \leq f(T)$ whenever $S \subseteq T$, we say that f is *monotone submodular.*

An equivalent definition, which better reflects the diminishing returns property, is given below.

Definition 2. For any sets $S, T \subseteq Y$ with $S \subseteq T$ and any $y \in Y \setminus T$, we have

$$f(S \cup \{y\}) - f(S) \geq f(T \cup \{y\}) - f(T). \tag{7}$$

Intuitively, the incremental increase of the function is larger when an element is added to a small set than to a larger set. In what follows, we will use the second definition.

Definition 3. Let $\mathcal{I}$ be a non-empty collection of subsets of a finite set Y. An ordered pair $\mathcal{M} = (Y, \mathcal{I})$, where $\mathcal{I} \subseteq 2^Y$, is called *independent* if, for all $B \in \mathcal{I}$, any set $A \subseteq B$ is also in $\mathcal{I}$. Furthermore, if for all $A \in \mathcal{I}$, $B \in \mathcal{I}$, $|A| < |B|$, there exists a $j \in B \setminus A$ such that $A \cup \{j\} \in \mathcal{I}$, then $\mathcal{M}$ is called a *matroid*. Moreover, $\mathcal{M} = (Y, \mathcal{I})$ is called *uniform matroid* if $\mathcal{I} = \{S \subseteq Y : |S| \leq N\}$.

The following theorem establishes the fact that the objective function $H(\mathbf{s})$ in (5) is monotone submodular, regardless of the obstacles that may be present in the mission space. This will allow us to apply results that quantify a solution obtained through the greedy algorithm relative to the global optimum in (5).

Theorem 1. *$H(\mathbf{s})$ defined in (5) is a monotone submodular set function.*

Proof. Let S and T, such that $S \subseteq T \subseteq F^D$, be two agent position vectors. Since $S \subseteq T$ and $0 \leq 1 - \hat{p}_i(x, s_i) \leq 1$ for any $s_i \in F^D$, we have

$$\prod_{s_i \in S}[1 - \hat{p}_i(x, s_i)] \geq \prod_{s_i \in T}[1 - \hat{p}_i(x, s_i)] \tag{8}$$

for all $x \in \Omega$. In addition, $H(S \cup \{s_k\})$ can be written as

$$\begin{aligned} &H(S \cup \{s_k\}) \\ &= \int_{\Omega} R(x) \left\{ 1 - [1 - \hat{p}_k(x, s_k)] \prod_{s_i \in S}[1 - \hat{p}_i(x, s_i)] \right\} dx \\ &= \int_{\Omega} R(x) \left\{ 1 - \prod_{s_i \in S}[1 - \hat{p}_i(x, s_i)] \right\} dx \\ &\quad + \int_{\Omega} R(x)\hat{p}_k(x, s_k) \prod_{s_i \in S}[1 - \hat{p}_i(x, s_i)]dx. \end{aligned}$$

The difference between $H(S)$ and $H(S \cup \{s_k\})$ is given by

$$\begin{aligned} &H(S \cup \{s_k\}) - H(S) \\ &= \int_{\Omega} R(x)\hat{p}_k(x, s_k) \prod_{s_i \in S}[1 - \hat{p}_i(x, s_i)]dx. \end{aligned} \tag{9}$$

Using the same derivation for T, we can obtain

$$\begin{aligned} &H(T\cup\{s_k\})-H(T) \\ &=\int_{\Omega} R(x)\hat{p}_k(x,s_k)\prod_{s_i\in T}[1-\hat{p}_i(x,s_i)]dx. \end{aligned} \tag{10}$$

From (9) and (10), the difference between $H(S\cup\{s_k\})-H(S)$ and $H(T\cup\{s_k\})-H(T)$ is

$$\begin{aligned} &[H(S\cup\{s_k\})-H(S)]-[H(T\cup\{s_k\})-H(T)] \\ &=\int_{\Omega} R(x)\hat{p}_k(x,s_k)\prod_{s_i\in S}[1-\hat{p}_i(x,s_i)]dx \\ &\quad -\int_{\Omega} R(x)\hat{p}_k(x,s_k)\prod_{s_i\in T}[1-\hat{p}_i(x,s_i)]dx. \end{aligned} \tag{11}$$

Using (8), it follows that the difference $[H(S\cup\{s_k\})-H(S)]-[H(T\cup\{s_k\})-H(T)]\geq 0$. Therefore, from Definition 2, $H(\mathbf{s})$ is submodular. Next, we prove that $H(\mathbf{s})$ is monotone, i.e., $H(S)\leq H(T)$. Subtracting $H(T)$ from $H(S)$ yields

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

Using (8), we have $H(S)-H(T)\leq 0$. Therefore, $H(\mathbf{s})$ is a monotone submodular function. ■

3.1.2. Lower bounds

We will use the definition

$$L(N)=\frac{f}{f^*}$$

from Section 1, where $f^\star$ is the global optimum of (5) and f is a feasible solution obtained by Algorithm 1 . Then, as shown in Nemhauser et al. (1978), a lower bound of $L(N)$ is $1-1/e$.

The lower bound of $L(N)$ can be improved by exploring the curvature information (total curvature and elemental curvature) of the objective function. The following results are based on the fact that $H(\mathbf{s})$ is a monotone submodular function satisfying $H(\emptyset)=0$ proved in Theorem 1 and $\mathcal{M}=(F^D,\mathcal{I})$ is a uniform matroid according to Definition 3.

Next, we consider the *total curvature*

$$c=\max_{j\in F^D}\left[1-\frac{H(F^D)-H(F^D\setminus j)}{H(\{j\})}\right] \tag{12}$$

introduced in Conforti and Cornuéjols (1984). Using c, the lower bound of $L(N)$ above is improved to be $T(c,N)$:

$$T(c,N)=\frac{1}{c}\left[1-(\frac{N-c}{N})^N\right], \tag{13}$$

where $c\in[0,1]$, and

$$T(c,N)\geq 1-\frac{1}{e}$$

for any $N\geq 1$ (Conforti & Cornuéjols, 1984). If $c=1$, the result is the same as the bound obtained in Nemhauser et al. (1978), Fisher, Nemhauser, and Wolsey (1978).

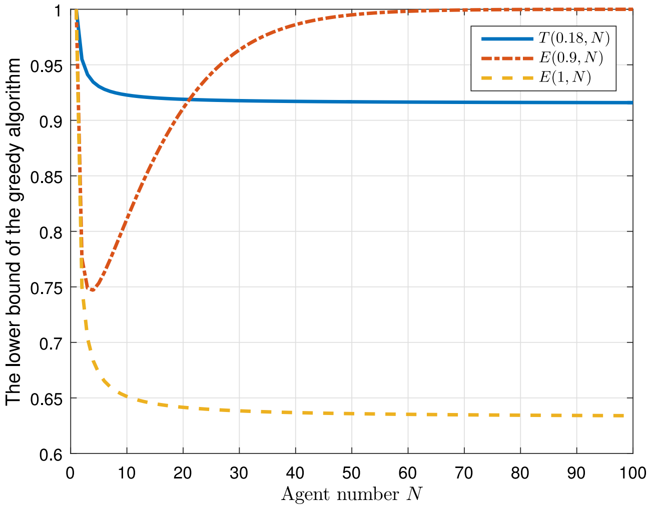


Fig. 4. $T(c,N)$ and $E(\alpha,N)$ as a function of the number of agents N.

In addition, we consider the *elemental curvature*

$$\alpha=\max_{S\subset F^D,i,j\in F^D\setminus S,i\neq j}\frac{H(S\cup\{i,j\})-H(S\cup\{j\})}{H(S\cup\{i\})-H(S)}, \tag{14}$$

based on which the following bound is obtained:

$$E(\alpha,N)=1-\left(\frac{\alpha+\cdots+\alpha^{N-1}}{1+\alpha+\cdots+\alpha^{N-1}}\right)^N \tag{15}$$

and it is shown in Wang et al. (2016) that $L(N)\geq E(\alpha,N)$. Note that $E(\alpha,N)$ can be simplified as follows:

$$E(\alpha,N)=\begin{cases}1-(\dfrac{N-1}{N})^N, & \text{when } \alpha=1;\\ 1-(\dfrac{\alpha-\alpha^N}{1-\alpha^N})^N, & \text{when } 0\leq\alpha<1.\end{cases} \tag{16}$$

Once $T(c,N)$ and $E(\alpha,N)$ are calculated, the larger one is the lower bound $L(N)$, defined as

$$L(N)=\max\{T(c,N),E(\alpha,N)\}. \tag{17}$$

Accordingly, we have $H(S)\geq L(N)H(S^*)$, where $S^\star$ is the global optimum set, and S is the set obtained by Algorithm 1 . Note that, in general, the computation of c is $O(n)$ and the computation of α is $O(n^2)$ where n is the cardinality of F^D. For the optimal coverage problem, we will provide a way to simplify the calculation of α to $O(n)$ in Eq. (23).

Fig. 4 shows the dependence of $T(c,N)$ and $E(\alpha,N)$ on the number of agents N for some specific values of c and α (as shown in the figure). Clearly, if $c<1$ and $\alpha<1$, then $L(N)$ in (17) is much tighter than $1-\frac{1}{e}$.

3.1.3. Curvature information calculation

We will derive the concrete form of the total curvature c and the elemental curvature α for the objective function $H(\mathbf{s})$ in (5). For notational convenience, $\hat{p}_i(x,s_i)$ is used without its arguments as long as this dependence is clear from the context.

Recall that F^D is the set of feasible agent positions. We can obtain from (5):

$$\begin{aligned} H(F^D)&=\int_{\Omega} R(x)\left[1-\prod_{i=1}^{n}(1-\hat{p}_i)\right]dx \\ &=\int_{\Omega} R(x)\left[1-(1-\hat{p}_j)\prod_{i=1,i\neq j}^{n}(1-\hat{p}_i)\right]dx, \end{aligned}$$

and

$$H(F^D \setminus \{s_j\}) = \int_{\Omega} R(x)\left[1 - \prod_{i=1, i\neq j}^{n}(1-\hat{p}_i)\right]dx.$$

The difference between $H(F^D)$ and $H(F^D \setminus \{s_j\})$ is

$$H(F^D) - H(F^D \setminus \{s_j\}) = \int_{\Omega} R(x)\hat{p}_j \prod_{i=1, i\neq j}^{n}[1-\hat{p}_i]dx. \tag{18}$$

When there is only one agent s_j, the objective function is

$$H(s_j) = \int_{\Omega} R(x)\hat{p}_j dx. \tag{19}$$

Combining (12), (18) and (19), we obtain

$$c = \max_{s_j \in F^D}\left[1 - \frac{\int_{\Omega} R(x)\hat{p}_j \prod_{i=1, i\neq j}^{n}[1-\hat{p}_i]dx}{\int_{\Omega} R(x)\hat{p}_j dx}\right]. \tag{20}$$

Remark 3. If the sensing capabilities of agents are weak, that is, $\hat{p}_i$ is small for most parts in the mission space, then $\prod_{i=1, i\neq j}^{n}(1-\hat{p}_i)$ is, in turn, close to 1, which leads to a small value of c. It follows from (13) that the lower bound $T(c, N)$ is a monotonically decreasing function of c and approaches 1 near $c = 0$. This implies that the solution of the greedy algorithm is very close to the global optimum when the sensing capabilities are weak.

Next, we calculate the elemental curvature α. From (9), the difference between $H(S)$ and $H(S \cup \{s_k\})$ is

$$H(S \cup \{s_k\}) - H(S) = \int_{\Omega} R(x)\hat{p}_k(x) \prod_{s_i \in S}[1-\hat{p}_i]dx. \tag{21}$$

Using the same derivation, we can obtain

$$\begin{aligned} &H(S \cup \{s_j, s_k\}) - H(S \cup \{s_j\}) \\ &\quad = \int_{\Omega} R(x)\hat{p}_k(1-\hat{p}_j) \prod_{s_i \in S}[1-\hat{p}_i]dx. \end{aligned} \tag{22}$$

The elemental curvature in (14) can then be calculated by

$$\begin{aligned} \alpha &= \max_{S, s_j, s_k} \frac{H(S \cup \{s_j, s_k\}) - H(S \cup \{s_j\})}{H(S \cup \{s_k\}) - H(S)} \\ &= \max_{S, s_j, s_k} \frac{\int_{\Omega} R(x)\hat{p}_k(1-\hat{p}_j) \prod_{s_i \in S}[1-\hat{p}_i]dx}{\int_{\Omega} R(x)\hat{p}_k \prod_{s_i \in S}[1-\hat{p}_i]dx} \\ &= \max_{S, s_j, s_k} 1 - \frac{\int_{\Omega} R(x)\hat{p}_k\hat{p}_j \prod_{s_i \in S}[1-\hat{p}_i]dx}{\int_{\Omega} R(x)\hat{p}_k \prod_{s_i \in S}[1-\hat{p}_i]dx} \\ &= 1 - \min_{s_j, x \in \Omega} \hat{p}_j(x, s_j). \end{aligned} \tag{23}$$

Remark 4. Observe that the elemental curvature turns out to be determined by a single agent. If there exists a pair (x, s_j) such that $x \in \bar{V}(s_j)$ in (2), then $\hat{p}_j(x, s_j) = 0$ and $\alpha = 1$. This may happen when there are obstacles in the mission space or the sensing capabilities of agents are weak (e.g., the sensing range is small or the sensing decay rate is large). On the other hand, if the sensing capabilities are so strong that $\hat{p}_j(x, s_j) \neq 0$ for any $x \in F, s_j \in F^D$, then $\alpha < 1$. In addition, $E(\alpha, N)$ is a monotonically decreasing function of α.

An interesting conclusion from this analysis is that $T(c, N)$ and $E(\alpha, N)$ are *complementary* with respect to the sensing capabilities of sensors. From Remark 3, $T(c, N)$ is large when the sensing capabilities are weak, while from Remark 4, $E(\alpha, N)$ is large when the sensing capabilities are strong. This conclusion is graphically depicted in Figs. 5 and 6 (where sensing capability varies from strong to weak). In Fig. 5, $E(\alpha, N)$ and $T(c, N)$ have been evaluated for $N = 10$ and $\delta = 80$ as a function of one of the measures of sensing capability, the sensing decay rate λ in (1), assuming all agents have the same sensing capabilities. One can see that for small values of λ, the bound $E(\alpha, 10)$ is close to 1 and dominates both $T(c, 10)$ and the well-known bound $1 - \frac{1}{e}$. Beyond a critical value of λ, it is $T(c, 10)$ that dominates and approaches 1 for large values of λ. Fig. 6 shows a similar behavior when $T(c, N)$ and $E(\alpha, N)$ are evaluated for $N = 10$ and $\lambda = 0.03$ as a function of the other measure of sensing capability, the sensing range δ. When the sensing range exceeds the distance of the diagonal of the mission space, there is no value in further increasing the sensing range and $E(\cdot)$ becomes constant. When $\delta > 20$, the sensing capabilities are strong and $T(\cdot)$ becomes constant. Therefore, both $E(\cdot)$ and $T(\cdot)$ become constant when δ exceeds corresponding thresholds. On the other hand, when the sensing range is smaller than some threshold, then $\alpha = 1$, and $E(1, 10) = 0.6513$.

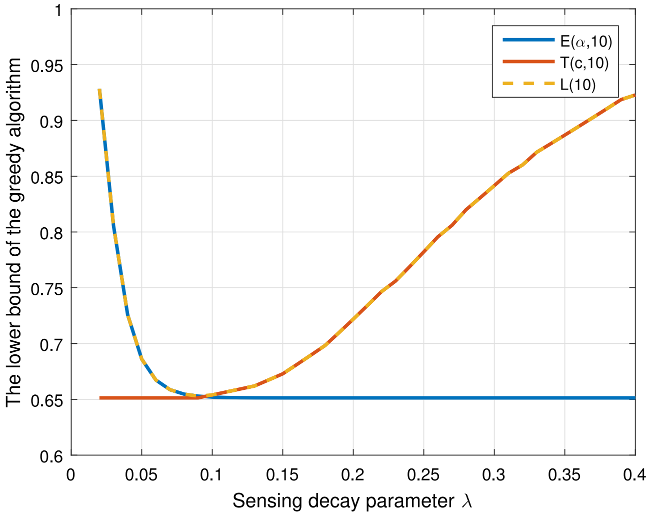


Fig. 5. Lower bound $L(10)$ as a function of the sensing decay rate of agents.

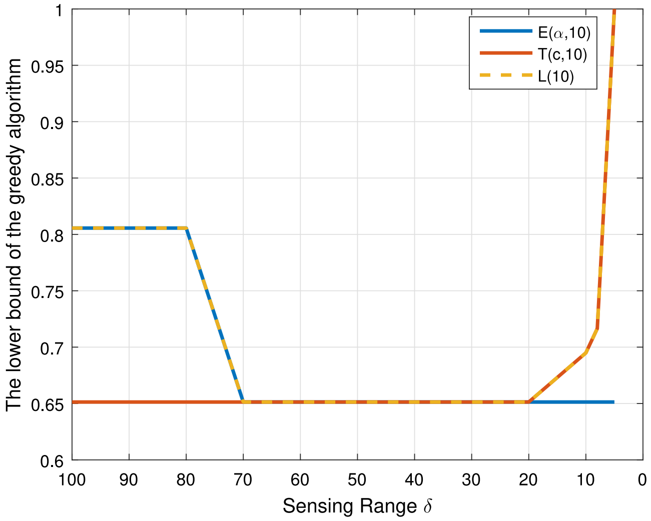


Fig. 6. Lower bound $L(10)$ as a function of the sensing range of agents.

Figs. 5 and 6 also illustrate the trade-off between the sensing capabilities and the coverage performance guarantee. Agents with strong capabilities obviously achieve better coverage performance. On the other hand, one can get a better guaranteed performance as the agents' capabilities get weaker. Therefore, if one is limited to agents with weak sensing capabilities in a particular setting, the use of $T(c, N)$ is appropriate and this trade-off may be exploited.

Remark 5. The following comments pertain to the accuracy of the greedy algorithm solution to the original optimal coverage problem (4). Finding a global optimal solution to the problem (5) by the brute-force search is unnecessary and it is adequate to be greedy with provable optimality bounds. The solution to (5) has no significance beyond the fact that it provides an initial condition for the original optimal coverage problem (4). On the other hand, the time complexity of the greedy algorithm increases linearly with the cardinality of the discrete set F^D. For this reason, it is not advisable to limit F^D to a small cardinality. We may get a tight optimality bound from the greedy algorithm for a small size F^D, but the gap between the optimal values of problems (4) and (5) is large.

3.2. Gradient algorithm

In this subsection, we use existing gradient-based algorithms with an initial deployment given by the greedy algorithm (Algorithm 1) shown in Algorithm 2 to seek better performance for the optimal coverage problem (4) in a continuous environment. Therefore, the gradient is calculated in the continuous environment, whereas the greedy algorithm is limited to a discrete environment. In particular, we use the distributed gradient-based algorithm developed in Zhong and Cassandras (2011):

$$s_i^{k+1} = s_i^k + \zeta_k \frac{\partial H(\mathbf{s})}{\partial s_i^k}, \quad k = 0, 1, \ldots \tag{24}$$

where the step size sequence $\{\zeta_k\}$ is appropriately selected to ensure convergence of the resulting trajectories for all agents (Bertsekas, 1995). The detailed calculation of $\frac{\partial H(\mathbf{s})}{\partial s_i^k}$ can be found in Sun et al. (2014). We only include all necessary notation and the final result of the partial derivative. The neighborhood set B_i of agent i is defined as:

$$B_i = \{k : \|s_i - s_k\| < 2\delta_i, \ k = 1, \ldots, N, \ k \neq i\} \tag{25}$$

This set includes all agents k whose sensing region Ω_k has a nonempty intersection with Ω_i, the sensing region of agent i. Let $\Phi_i(x) = \prod_{k \in B_i}[1 - \hat{p}_k(x, s_k)]$ denote the joint probability that a point $x \in \Omega$ is not detected by any neighbor agent of agent i.

Let v be a reflex vertex of an obstacle and $x \in F$ a point visible from v. A set of points in a ray starting from v and extending in the direction of $v - x$, is defined by $I(v, x) = \{q \in V(v) : q = \lambda v + (1 - \lambda)x, \lambda > 1\}$. The ray intersects the boundary of F at an *impact* point. The segment from v to the impact point is $I(v, x)$.

An *anchor* of s_i is a reflex vertex v such that it is visible from s_i and $I(v, s_i)$ is not empty. Denote the anchors of s_i by v_{ij}, $j = 1, \ldots, Q(s_i)$, where $Q(s_i)$ is the number of anchors of s_i. An *impact* point of v_{ij}, denoted by V_{ij}, is the intersection of $I(v_{ij}, s_i)$ and ∂F. As an example, in Fig. 1, v_{i1}, v_{i2}, v_{i3} are anchors of s_i, and V_{i1}, V_{i2}, V_{i3} are the corresponding impact points. Let $D_{ij} = \|s_i - v_{ij}\|$ and $d_{ij} = \|V_{ij} - v_{ij}\|$. Define $\theta_{ij} = \arctan \frac{|s_i - v_{ij}|_y}{|s_i - v_{ij}|_x}$ which satisfies $\theta_{ij} \in [0, \pi/2]$ as the angle formed by $s_i - v_{ij}$ and the x-axis, $\Gamma_i = \{j : D_{ij} < \delta_i, j = 1, \ldots, Q(s_i)\}$, $z_{ij} = \min(d_{ij}, \delta_i - D_{ij})$ and $\rho_{ij}(r)\rho_{ij}(r) = (V_{ij} - v_{ij})\frac{r}{d_{ij}} + v_{ij}$ is the Cartesian coordinate of a point on I_{ij} which is at a distance r from v_{ij}.

With these notation, the final result of the partial derivative is

$$\frac{\partial H_i(\mathbf{s})}{\partial s_{ix}} = \int_{V(s_i)} w_1(x, s_i)\frac{(x - s_i)_x}{d_i(x)}dx + \sum_{j \in \Gamma_i} sgn(n_{jx})\frac{\sin\theta_{ij}}{D_{ij}}\int_0^{z_{ij}} w_2(\rho_{ij}(r), s_i)rdr \tag{26}$$

$$\frac{\partial H_i(\mathbf{s})}{\partial s_{iy}} = \int_{V(s_i)} w_1(x, s_i)\frac{(x - s_i)_y}{d_i(x)}dx + \sum_{j \in \Gamma_i} sgn(n_{jy})\frac{\cos\theta_{ij}}{D_{ij}}\int_0^{z_{ij}} w_2(\rho_{ij}(r), s_i)rdr \tag{27}$$

where $w_1(x, s_i) = -R(x)\Phi_i(x)\frac{dp_i(x,s_i)}{dd_i(x)}$ controls the mechanism through which agent i is attracted to different points $x \in V(s_i)$ through $\frac{x - s_i}{d_i(x)}$, $w_2(x, s_i)$ in the second integral controls the attraction that boundary points exert on node i with the geometrical features of the mission space contributing through n_{jx}, n_{jy}, θ_{ij}, and D_{ij} (Sun et al., 2014).

As discussed in Zhong and Cassandras (2011), we assume that the agent locations do not coincide with a reflex vertex, a polygonal inflection, or a bitangent, at which points $H(\mathbf{s})$ is generally not differentiable. To take these points into account, one can replace the standard gradient-based algorithm in (24) by subgradient algorithms or bundle methods which aggregate the subgradient information in past iterations; see Zhong and Cassandras (2011) for details.

Algorithm 2 *Greedy-Gradient Algorithm*

```
Input: Objective function $H(\mathbf{s})$
Output: Agent positions $\mathbf{s}$
Initialization: $\mathbf{s}$ given by Greedy Algorithm 1
while the stopping criterion is not satisfied do
    Choose a step size $\zeta > 0$
    for $i = 1, \ldots, N$ do
        Determine a searching direction $\frac{\partial H(\mathbf{s})}{\partial s_i}$
        Update: $s_i \leftarrow s_i + \zeta\frac{\partial H(\mathbf{s})}{\partial s_i}$
    end for
end while
return s
```

The stopping criterion is of the form $\|\frac{\partial H(\mathbf{s})}{\partial s_i}\| \leq \eta$, where η is a small positive scalar.

4. Simulation results

In this section, we illustrate through simulation our analysis and the use of the GGA (Algorithm 2), and show the performance improvements obtained when compared with the greedy algorithm (Algorithm 1) and the distributed gradient algorithm in (24) for coverage problems in a variety of mission spaces (no obstacles, a wall-like obstacle, a maze-like obstacle, a collection of random obstacles, and a building with multiple rooms). The mission space is a 60 × 50 rectangular area. All agents have the same sensing range $\delta_i = 80$, $i = 1, \ldots, N$. For each mission space, two different sensing decay rates λ are used for the comparison as shown in Figs. 8–9, 10–11, 15–16, 17–18 and 19–20, where (a) shows the results of our distributed gradient-based algorithm with initialization at the left upper corner, (b) shows the results under the greedy algorithm, and (c) shows the results under the GGA. The mission space is colored from dark to light as the joint detection probability (our objective function) decreases: the joint detection probability is ≥ 0.97 for purple areas, ≥ 0.50 for green areas, and near zero for white areas.

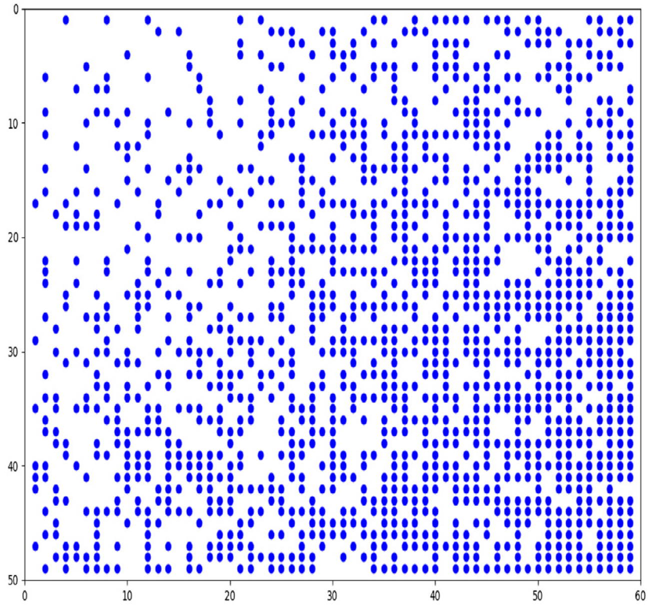


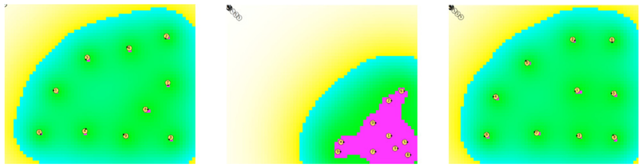
Fig. 7. F^D for $R(x) = \frac{(x_x+x_y)}{(x_{max}+y_{max})}$ with $\Delta = 1$.

When applying the greedy algorithm, the continuous mission space F is discretized to F^D by the method described in Section 3.1 with $\Delta = 1$. For the cases in Figs. 8–9, $R(x)$ is a linear function of $x = [x_x, x_y]$ with the form $R(x) = \frac{x_x+x_y}{\max\{x_x\}+\max\{x_y\}}$. The set F^D is shown Fig. 7. As we discussed in Section 3.1, more points are selected around the right bottom corner with high event density $R(x)$ than the left upper corner with low event density in F^D. For all remaining cases, there are obstacles and $R(x) = 1$ for $x \in F$, which facilitates comparison with our previous work (Sun et al., 2014). When $R(x) = 1$, all intersecting points are selected in F^D except those that lie within obstacles.

Figs. 8–9 show that the GGA algorithm is capable of dealing with the non-uniform distribution $R(x)$. Agents cluster around the right bottom corner where points have high event density $R(x)$, as expected. The GGA algorithm and the gradient-based algorithm perform similarly, while the greedy algorithm, by its very nature, performs poorly.

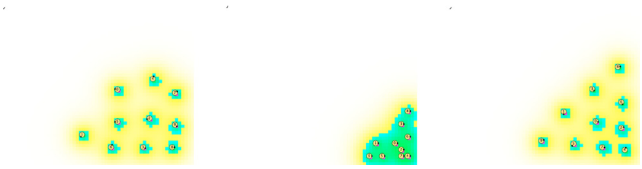
For cases with obstacles in the mission space, we first discuss in detail and provide insights for the wall-like mission space case and then include some additional interesting cases.

Fig. 12 depicts the evolution of the coverage function $H(\mathbf{s})$ when the GGA is applied to the case in Fig. 10. During the first 10 steps, only the greedy algorithm is in operation and the coverage function evolves as the 10 agents are added one by one, illustrating the submodularity property. Subsequently, the gradient-based algorithm is set into operation and the objective function value is further increased from 1813.3 to 1846.3. The left upper corner initialization is intentionally a very poor choice for a wall-like mission space as seen from Figs. 10a and 11a. This choice and the weak sensing ability of the agents in these particular examples result in poor coverage as the agents have no advance knowledge of the large uncovered space and they are limited to covering only the left area in the mission space. For a fair comparison, we apply random initialization twice to the case in Fig. 10a, where the initial agents' locations are shown in Figs. 13a and 14a, respectively. The corresponding coverage performance for the agents' final locations shown in Figs. 13b and 14b is $H(\mathbf{s}) = 1472.8$ and $H(\mathbf{s}) = 1670.9$, respectively. The coverage performance with random initialization is not better than the one with the initial locations generated by the greedy algorithm as shown in Fig. 10c.



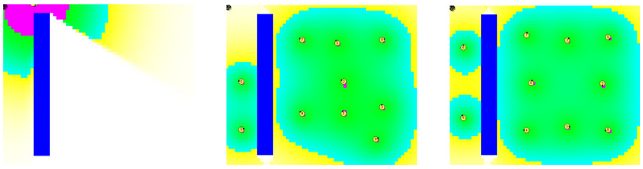
(a) $H(s) = 1077.8$ (b) $H(s) = 783.2$ (c) $H(s) = 1078.0$

Fig. 8. The decay factor $\lambda = 0.12$, and no obstacles in the mission space.



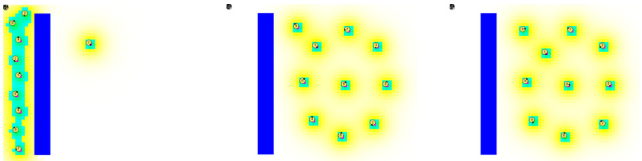
(a) $H(\mathbf{s}) = 242.6$ (b) $H(\mathbf{s}) = 174.6$ (c) $H(\mathbf{s}) = 244.0$

Fig. 9. The decay factor $\lambda = 0.4$, and no obstacles in the mission space.



(a) $H(\mathbf{s}) = 437.1$ (b) $H(\mathbf{s}) = 1813.3$ (c) $H(\mathbf{s}) = 1846.3$

Fig. 10. The decay factor $\lambda = 0.12$, and a wall-like obstacle in the mission space.



(a) $H(\mathbf{s}) = 269.6$ (b) $H(\mathbf{s}) = 371.9$ (c) $H(\mathbf{s}) = 373.2$

Fig. 11. The decay factor $\lambda = 0.4$, and a wall-like obstacle in the mission space.

In the rest of the examples, such as the maze-like obstacle, a collection of random obstacles, and a building with multiple rooms, the greedy algorithm and the GGA outperform the basic gradient-based algorithm as well. Moreover, the results of the GGA significantly improve upon those reported in our previous work (Sun et al., 2014). As an example, in the cases of Fig. 19 with $\lambda = 0.12$, the objective function value is improved from a value of 1419.5 reported in Sun et al. (2014) (using the distributed gradient-based algorithm with improvements provided through the use of boosting functions) to 1466.9.

5. Conclusions and future work

We have obtained an initial solution to the optimal coverage problem by using the greedy algorithm to approximate the

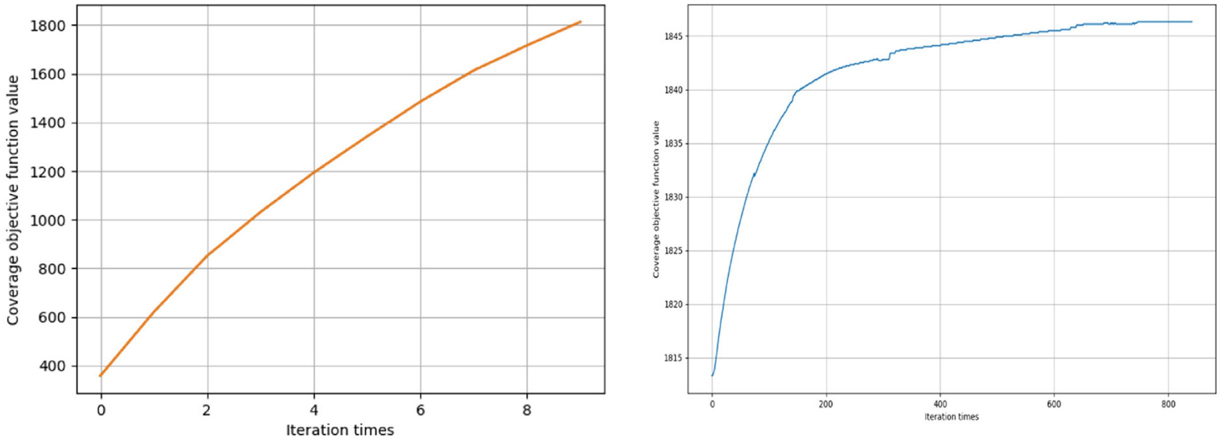


Fig. 12. Evolution of $H(\mathbf{s})$ when the GGA is applied: first, only the greedy algorithm is applied, then (after 10 iterations), the gradient-based algorithm is applied.

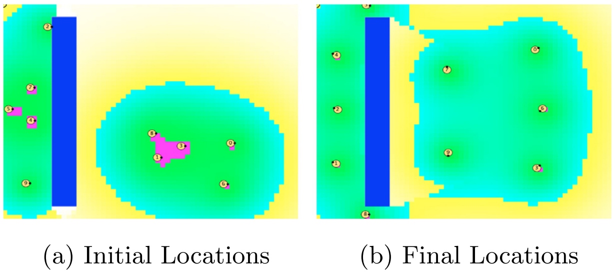

(a) Initial Locations (b) Final Locations

Fig. 13. Gradient-based algorithm with random initialization to the case in Fig. 10(a).

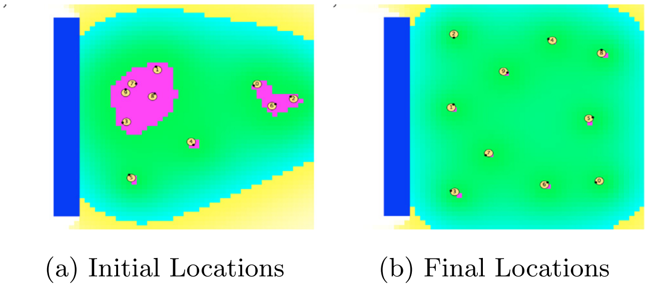

(a) Initial Locations (b) Final Locations

Fig. 14. Gradient-based algorithm with random initialization to the case in Fig. 10(a).

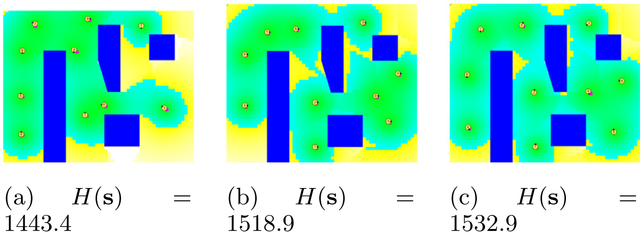

(a) $H(\mathbf{s}) = 1443.4$ (b) $H(\mathbf{s}) = 1518.9$ (c) $H(\mathbf{s}) = 1532.9$

Fig. 15. The decay factor $\lambda = 0.12$, in a general mission space.

solution of non-convex optimal coverage problems. The obtained greedy solution ensures a guaranteed lower bound relative to the global optimum of the approximated optimal coverage problem which is significantly tighter than the one well-known in the

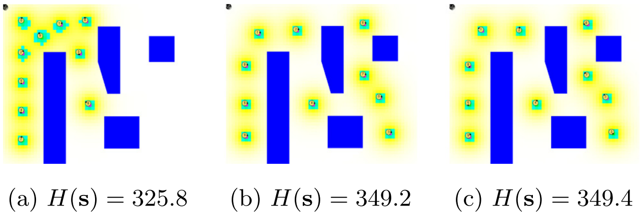

(a) $H(\mathbf{s}) = 325.8$ (b) $H(\mathbf{s}) = 349.2$ (c) $H(\mathbf{s}) = 349.4$

Fig. 16. The decay factor $\lambda = 0.4$, in a general mission space.

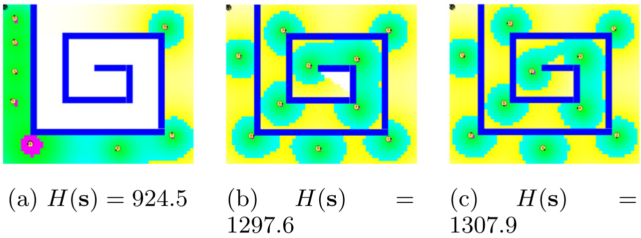

(a) $H(\mathbf{s}) = 924.5$ (b) $H(\mathbf{s}) = 1297.6$ (c) $H(\mathbf{s}) = 1307.9$

Fig. 17. The decay factor $\lambda = 0.12$, in a maze mission space.

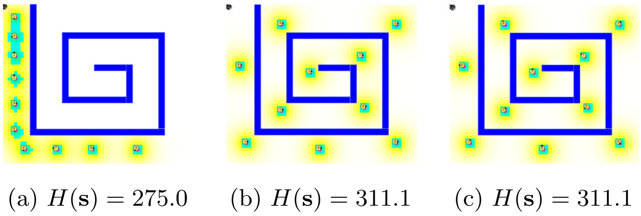

(a) $H(\mathbf{s}) = 275.0$ (b) $H(\mathbf{s}) = 311.1$ (c) $H(\mathbf{s}) = 311.1$

Fig. 18. The decay factor $\lambda = 0.4$, in a maze mission space.

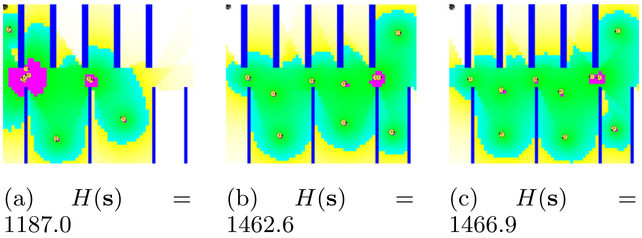

(a) $H(\mathbf{s}) = 1187.0$ (b) $H(\mathbf{s}) = 1462.6$ (c) $H(\mathbf{s}) = 1466.9$

Fig. 19. The decay factor $\lambda = 0.12$, in a room mission space.

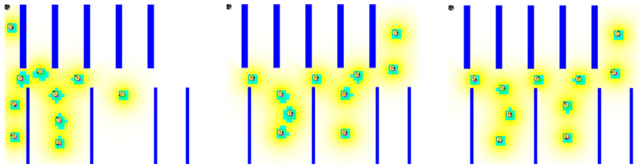

(a) $H(\mathbf{s}) = 303.1$ (b) $H(\mathbf{s}) = 344.5$ (c) $H(\mathbf{s}) = 347.2$

Fig. 20. The decay factor $\lambda = 0.4$, in a room mission space.

literature to be $1 - 1/e$. This is made possible by proving that our coverage metric of the approximated optimal coverage problem is monotone submodular and by calculating its total curvature and its elemental curvature. Therefore, we are able to reduce the theoretical performance gap between optimal and suboptimal solutions enabled by the submodularity theory. Moreover, we have shown that the two new bounds derived are complementary with respect to the sensing capabilities of the agents and each one approaches its maximal value of 1 under different conditions on the sensing capabilities, enabling us to select the most appropriate one depending on the characteristics of the agents at our disposal. In addition, by combining the greedy algorithm with a distributed gradient-based algorithm we have proposed a greedy-gradient algorithm (GGA) so as to improve the coverage performance by searching in a continuous feasible region with initial conditions provided by the greedy algorithm. We have included simulation results uniformly showing that the proposed distributed GGA outperforms other related methods we are aware of.

An interesting future research direction is to study whether a distributed greedy algorithm can be developed and whether the lower bounds obtained through the associated curvatures are still as tight as those we have obtained so far.

References

Berman, O., & Krass, D. (2002). The generalized maximal covering location problem. *Computers & Operations Research*, *29*(6), 563–581.

Bertsekas, D. P. (1995). *Nonlinear programming*. Athena Scientific.

Bertsimas, D., & Tsitsiklis, J. (1993). Simulated annealing. *Statistical Science*, 10–15.

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 *Proc. IEEE int. conf. on robotics and automation* (pp. 4982–4989).

Caicedo-Nuez, C., & Zefran, M. (2008). A coverage algorithm for a class of non-convex regions. In *Proc. IEEE conf. on decision and control* (pp. 4244–4249).

Caicedo-Nunez, C. H., & Zefran, M. (2008). Performing coverage on nonconvex domains. In *Proc. IEEE conf. on control applic.* (pp. 1019–1024).

Cassandras, C. G., & Sensor, W. Li. (2005). networks and cooperative control. *European Journal of Control*, *11*(4–5), 436–463.

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.

Fisher, M. L., Nemhauser, G. L., & Wolsey, L. A. (1978). An analysis of approximations for maximizing submodular set functions —II. In *Polyhedral combinatorics* (pp. 73–87). Springer.

Gusrialdi, A., Hirche, S., Hatanaka, T., & Fujita, M. (2008). Voronoi based coverage control with anisotropic sensors. In *Proc. amer. control conf.* (pp. 736–741).

Gusrialdi, A., & Zeng, L. (2011). Distributed deployment algorithms for robotic visual sensor networks in non-convex environment. In *Proc. IEEE int. conf. on networking. sensing and control* (pp. 445–450).

Kantaros, Y., Thanou, M., & Tzes, A. (2015). Distributed coverage control for concave areas by a heterogeneous robot-swarm with visibility sensing constraints. *Automatica*, *53*, 195–207.

Khuller, S., Moss, A., & Naor, J. S. (1999). The budgeted maximum coverage problem. *Information Processing Letters*, *70*(1), 39–45.

Krause, A., Leskovec, J., Guestrin, C., VanBriesen, J., & Faloutsos, C. (2008). Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, *134*(6), 516–526.

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

Liu, Y., Zhang, Z., Chong, E. K. P., & Pezeshki, A. (2016). Performance bounds for the k-batch greedy strategy in optimization problems with curvature. In *2016 American control conference* (pp. 7177–7182).

Marier, J. S., Rabbath, C. A., & Léchevin, N. (2012). Visibility-limited coverage control using nonsmooth optimization. In *2012 American control conference* (pp. 6029–6034).

Meguerdichian, S., Koushanfar, F., Potkonjak, M., & Srivastava, M. B. (2001). Coverage problems in wireless ad-hoc sensor networks. In *Proc. 20th joint conf. of the IEEE computer and commun. societies, Vol. 3* (pp. 1380–1387).

Nemhauser, G. L., Wolsey, L. A., & Fisher, M. L. (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 *Proc. IEEE int. 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 *Proc. IEEE conf. on decision and control* (pp. 3701–3706).

Sun, X., Cassandras, C. G., & Meng, X. (2017). A submodularity-based approach for multi-agent optimal coverage problems. In *Proc. IEEE conf. on decision and control* (pp. 4082–4087).

Van Laarhoven, P. J., & Aarts, E. H. (1987). *Simulated annealing*. Springer.

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.

Zhang, Z., Chong, E. K., Pezeshki, A., & Moran, W. (2016). String submodular functions with curvature constraints. *IEEE Transactions on Automatic Control*, *61*(3), 601–616.

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.

Xinmiao Sun received the B.E. degree in Automation from Beijing Institute of Technology, Beijing, China, in 2012, and Ph.D. degree in Systems Engineering from Boston University, in 2017.

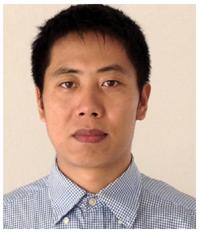
Her research interests include optimization and control of multi-agent systems, anomaly detection for discrete manufacturing systems, and simulation-based optimization.

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 co-founder of Boston University's Center for Information and Systems Engineering (CISE). He received degrees from Yale University (B.S., 1977), Stanford University (M.S.E.E., 1978), and Harvard University (S.M., 1979; Ph.D., 1982). 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 simulation, with applications to computer and sensor networks, manufacturing systems, and transportation systems. He has published about 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 Congres*s, the *American Control Conference* in 2001 and the *IEEE Conference on Decision and Control* in 2002 and 2016, and has also been an IEEE Distinguished Lecturer.

He is the recipient of several awards, including the 2011 IEEE Control Systems Technology Award, the Distinguished Member Award of the IEEE Control Systems Society (2006), 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 (for a "Smart Parkin" system and for the analytical engine of the Street Bump system respectively), 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.



Xiangyu Meng received his Ph.D. degree in Control Systems from University of Alberta, Canada, in 2014. He was a Research Associate in the Department of Mechanical Engineering at the University of Hong Kong between June 2007 and July 2007, and between November 2007 and January 2008. He was a Research Award Recipient in the Department of Electrical and Computer Engineering at the University of Alberta between February 2009 and August 2010. In December 2014, he joined the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, as a Research Fellow. Since January 2017, he has been with the Division of Systems Engineering at the Boston University, United States, where he is a Postdoctoral Associate. His research interests include smart cities, connected and automated vehicles, and cyber–physical systems.