Automatica 144 (2022) 110493

Contents lists available at ScienceDirect

Automatica

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

A new performance bound for submodular maximization problems and its application to multi-agent optimal coverage problems[☆]

Shirantha Welikala[a,b,*], Christos G. Cassandras[a], Hai Lin[b], Panos J. Antsaklis[b]

[a] Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA 02446, USA
[b] Department of Electrical Engineering, University of Notre Dame, South Bend, IN 02446, USA

ARTICLE INFO

Article history:
Received 5 October 2021
Received in revised form 15 March 2022
Accepted 11 May 2022
Available online xxxx

Keywords:
Multi-agent systems
Optimization
Cooperative control
Control of networks
Persistent monitoring
Parametric control

ABSTRACT

Several important problems in multi-agent systems, machine learning, data mining, scheduling and others, may be formulated as set function maximization problems subject to cardinality constraints. In such problems, the set (objective) functions of interest often have *monotonicity* and *submodularity* properties. Hence, the class of monotone submodular set function maximization problems has been widely studied in the literature. Owing to its challenging nature, almost all existing solutions for this class of problems are based on greedy algorithms. A seminal work on this topic has exploited the submodularity property to prove a (1-1/e) performance bound for such greedy solutions. More recent literature on this topic has been focused on exploiting different *curvature* properties to establish improved (tighter) performance bounds. However, such improvements come at the cost of enforcing additional assumptions and increasing computational complexity while facing significant inherent limitations. In this paper, first, a brief review of existing performance bounds is provided. Then, a new performance bound that does not require any additional assumptions and is both practical and computationally inexpensive is proposed. In particular, this new performance bound is established based on a series of upper bounds derived for the objective function that can be computed in parallel with the execution of the greedy algorithm. Finally, to highlight the effectiveness of the proposed performance bound, extensive numerical results obtained from a well-known class of multi-agent coverage problems are provided.

1. Introduction

The salient feature that characterizes a *submodular* set function is its *diminishing returns* property. Simply, this property means that the marginal gain (return) of adding an element to a set decreases as the set grows (accumulates new elements). In this sense, submodularity bears a similarity to concavity. However, it has been established that maximizing submodular set functions is NP-hard (Corneuejols, Fisher, & Nemhauser, 1977; Nemhauser, Wolsey, & Fisher, 1978) while minimizing them can be achieved in polynomial time (Grötschel, Lovász, & Schrijver, 1981; Schrijver, 2000). Therefore, submodularity also has a resemblance to convexity. Despite this duality, submodular set functions appear naturally in many real-world problems such as in the coverage control (Sun, Cassandras, & Meng, 2019; Sun, Welikala, & Cassandras, 2020), persistent monitoring (Rezazadeh & Kia, 2019), feature selection (Das & Kempe, 2008), document summarization (Lin & Bilmes, 2011), image segmentation (Jegelka & Bilmes, 2011), marketing (Kempe, Kleinberg, & Tardos, 2003), data mining (Mirzasoleiman, Karbasi, Sarkar, & Krause, 2013), machine scheduling (Liu, 2020) and recommender systems (El-Arini & Guestrin, 2011).

Motivated by its applicability, submodular maximization problems have been theoretically studied in the literature under a diverse set of conditions (Liu, Chong, Pezeshki, & Zhang, 2020). For example, the submodular *objective function* is assumed to be: monotone in Wang, Moran, Wang, and Pan (2016), non-monotone in Fahrbach, Mirrokni, and Zadimoghaddam (2019) and weakly submodular in Khanna, Elenberg, Dimakis, Negahban, and Ghosh (2017). Similarly, different types of set variable *constraints* such as cardinality (Nemhauser et al., 1978), matroid (Fisher, Nemhauser, & Wolsey, 1978), knapsack (Wolsey, 1982) and matchoid (Badanidiyuru, Karbasi, Kazemi, & Vondrak,

☆ This work was supported in part by NSF under grants ECCS-1931600, DMS-1664644, CNS-1645681, CNS-1830335 and IIS-2007949, by AFOSR under grant FA9550-19-1-0158, by ARPA-E under grant DE-AR0001282 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 Sergio Grammatico under the direction of Editor Ian R. Petersen.

* Corresponding author at: Department of Electrical Engineering, University of Notre Dame, South Bend, IN 02446, USA.

E-mail addresses: wwelikal@nd.edu (S. Welikala), cgc@bu.edu (C.G. Cassandras), hlin1@nd.edu (H. Lin), pantsakl@nd.edu (P.J. Antsaklis).

https://doi.org/10.1016/j.automatica.2022.110493

2020) have been considered throughout the literature. Even though submodular maximization problems have been studied under various conditions, their solutions are predominantly based on greedy algorithms. In this paper, similar to Conforti and Cornuéjols (1984), Liu, Chong, and Pezeshki (2018), Nemhauser et al. (1978) and Wang et al. (2016), we consider the class of submodular maximization problems where the objective function is monotone, the set variable is cardinality constrained, and the solution is obtained by a vanilla greedy algorithm. Such maximization problems arise naturally from applications like coverage control (Sun et al., 2019, 2020), persistent monitoring (Rezazadeh & Kia, 2019), machine scheduling (Liu, 2020) and resource allocation (Liu et al., 2020). However, in contrast to the prior work in Conforti and Cornuéjols (1984), Liu et al. (2018), Nemhauser et al. (1978) and Wang et al. (2016), here we propose a novel *performance bound* for the obtained greedy solutions.

Formally, a performance bound of a greedy solution is a lower bound to the ratio f^G/f^* so that $\beta \leq f^G/f^*$, where f^G and f^* correspond to the objective function values under the greedy solution and the global optimal solution, respectively. For monotone submodular objective functions, the seminal papers Fisher et al. (1978) and Nemhauser et al. (1978) respectively show that $\beta = \frac{1}{2}$ when the set variable is constrained over a general matroid and $\beta = (1-(1-\frac{1}{N})^N)$ when the set variable's cardinality is constrained by N. Note that having a performance bound closer to 1 is preferred, as it implies that the greedy solution is almost globally optimal.

Recent work on this class of problems has shown an increasing interest in improving the aforementioned conventional performance bounds by exploiting structural properties of the underlying problem. The typical approach is first to define a *curvature* measure that characterizes the structural properties of the underlying objective function, the feasible space and the generated greedy solution. Then, based on this curvature measure, an improved (closer to 1 compared to conventional counterparts) performance bound is established. For example, Conforti and Cornuéjols (1984) defined a curvature measure named *total curvature* based on the nature of the objective function and the feasible space. Then, a provably improved performance bound was developed using the said total curvature measure. The authors of Conforti and Cornuéjols (1984) also proposed another curvature metric named *greedy curvature* based on the generated greedy solution and used it to develop another performance bound. The same procedure was followed in Liu et al. (2018) and Wang et al. (2016) to propose two new curvature metrics named *elemental curvature* and *partial curvature* respectively and then to develop corresponding performance bounds.

In this work, we first review the aforementioned total, greedy, elemental and partial curvature measures proposed in Conforti and Cornuéjols (1984), Liu et al. (2018) and Wang et al. (2016) while outlining their strengths and weaknesses. In particular, we point out that some of these curvature measures can be: (i) computationally expensive to obtain, (ii) require enforcing additional assumptions and (iii) may have inherent limitations that prevent them from providing improved performance bounds (e.g., if the submodularity property of the objective function is strong or weak). We next propose a novel curvature measure (which we named the *extended greedy curvature*) along with a corresponding performance bound that can be computed efficiently in parallel with the execution of the greedy algorithm. We show that this performance bound may be improved by executing extra greedy iterations (hence the name "extended"). This new performance bound does not require any additional assumptions, and it also does not suffer from the said inherent limitations of its predecessors. Finally, we use a widely studied class of multi-agent coverage problems (Sun et al., 2019, 2020) and implement all the aforementioned performance bounds to highlight the effectiveness of the proposed performance bound in this paper.

The paper is organized as follows. The used preliminary concepts and notations are introduced in Section 2. A brief review of existing performance bounds is provided in Section 3. Section 4 presents the details of the proposed new performance bound. The multi-agent coverage problem setup and the observed numerical results are reported in Section 5 before concluding the paper in Section 6.

2. Preliminaries

We consider $X = \{x_1, x_2, \dots, x_M\}$ to be the finite *ground set* that represents all possible options/actions available. The *set function* $f : 2^X \to \mathbb{R}_{\geq 0}$ is considered as the objective function where 2^X denotes the *power set* of X. We use the notation

$$\Delta f(x|A) \triangleq f(A \cup \{x\}) - f(A), \tag{1}$$

to represent the *marginal gain* value of adding an element $x \in X \backslash A$ to the set $A \subset X$ (where "$\cdot\backslash\cdot$" stands for the set subtraction operation). Note that this notation can also be used more liberally as $\Delta f(B|A) \triangleq f(A \cup B) - f(A)$ for any $A, B \subseteq X$ (here the set $B \subseteq X$ is allowed to be such that $A \cap B \neq \emptyset$).

Definition 1. Over the ground set X, the set function f is:

(a) *normalized* if $f(\emptyset) = 0$,
(b) *monotone* if $f(B) \leq f(A)$ for all B, A where $B \subseteq A \subseteq X$,
(c) *submodular* if $\Delta f(x|A) \leq \Delta f(x|B)$ for all x, A, B where $B \subseteq A \subseteq X$ and $x \in X\backslash A$, or equivalently, if $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$ for all $A, B \subseteq X$,
(d) a *polymatroid* set function (Liu et al., 2018) if it is normalized, monotone and submodular.

Note that the first equivalent condition given for the submodularity property (in Definition 1(c)) is more commonly known as the *diminishing returns* condition. The following lemma presents a preliminary result regarding the marginal gain function that will be exploited in the sequel.

Lemma 1. *If $f(Y)$ is a polymatroid set function over the ground set X, for a fixed set $A \subset X$, the set function $g(Y) \triangleq \Delta f(Y|A)$ over the set $X\backslash A$ is also a polymatroid set function.*

Proof. Since $g(\emptyset) = \Delta f(\emptyset|A) = f(A \cup \emptyset) - f(A) = 0$, $g(Y)$ is normalized. According to the definition of $g(Y)$ and (1), for any set $B \in X\backslash A$, $g(B) = \Delta f(B|A) = f(B \cup A) - f(A)$. Similarly, for any set $C \in X\backslash A$, $g(C) = f(C \cup A) - f(A)$. Now, in order for $g(Y)$ to be monotone, according to Definition 1(b), for any $B \subseteq C \subseteq (X\backslash A)$, $g(B) \leq g(A)$, i.e., $f(B \cup A) \leq f(C \cup A)$. Note that the latter inequality holds true as f is monotone over the set X and $(B \cup A) \subseteq (C \cup A) \subseteq X$. Hence, $g(Y)$ is monotone over the set $X\backslash A$. Similarly, the inequality condition given in Definition 1(c) can be established for $g(Y)$ to prove its submodularity. Hence, $g(Y)$ is a polymatroid set function over the set $X\backslash A$.

Submodular maximization problem. Recall that the domain of the objective function $f(Y)$ is the power set 2^X (by definition). However, depending on the application, the number of options that can be selected from the ground set X to form the set variable Y may be limited. To this end, we assume that only N options can be selected from X, where $1 < N < M = |X|$ and "$|\cdot|$" represents the cardinality operator. Formally, this constraint on Y is represented by writing $Y \in \mathscr{I}^N$ where $\mathscr{I}^N \triangleq \{Y : Y \subseteq X, |Y| \leq N\}$. We also point out that the set system $(X, \mathscr{I}^N)$ is commonly known in the literature (Liu et al., 2018) as a *uniform* *matroid* of rank N. We now can state the considered submodular maximization problem in this paper as follows.

For a given *polymatroid* set function $f : 2^X \to \mathbb{R}_{\geq 0}$ over the *uniform matroid* $(X, \mathscr{I}^N)$ find the optimal set Y^* where

$$Y^* = \arg\max_{Y \in \mathscr{I}^N} f(Y). \tag{2}$$

The above problem is an NP-hard combinatorial optimization problem that has been widely studied in the literature (Liu et al., 2018). One trivial approach to solve (2) is to use a brute force search algorithm that evaluates f over each element in $\mathscr{I}^N$. While such an approach can give the exact global optimal Y^*, in many applications of interest, it is computationally intractable due to the involved search space size: $|\mathscr{I}^N| = \sum_{r=0}^{N} \frac{M!}{r!(M-r)!}$, which is of complexity $O(M!)$.

The greedy solution. As an alternative, a widely popular computationally efficient approach to generate a reasonable approximate (sub-optimal) solution to (2) is to use a *greedy algorithm*. A typical greedy algorithm (considered in this paper as well as in many others, including Conforti & Cornuéjols, 1984; Liu et al., 2018; Nemhauser et al., 1978; Wang et al., 2016) is given in Algorithm 1. In the remainder of this paper, the solution generated by this greedy algorithm is referred to as the *greedy solution* and is denoted by Y^G.

Algorithm 1 The greedy algorithm to solve (2)

1: $i = 0$; $Z^i = \emptyset$; ▷ Greedy iteration index and solution
2: **for** $i = 1, 2, 3, \ldots, N$ **do**
3: $\quad z^i = \arg\max_{\{x: Z^{i-1} \cup \{x\} \in \mathscr{I}^N\}} \Delta f(x|Z^{i-1})$; ▷ New option
4: $\quad Z^i = Z^{i-1} \cup \{z^i\}$; ▷ Append the new option
5: **end for**
6: $Y^G := Z^N$; **Return** Y^G;

In practice, the performance of the greedy solution is usually sub-optimal (i.e., $f(Y^G) \leq f(Y^*)$). However, its proximity to the global optimal performance can be characterized using the concept of *performance bound* defined next.

Definition 2. A valid *performance bound* (denoted by β) for the greedy solution Y_G obtained for the problem (2) is a theoretically established lower bound for the ratio $\frac{f(Y^G)}{f(Y^*)}$, i.e.,

$$\beta \leq \frac{f(Y^G)}{f(Y^*)}. \tag{3}$$

Note that if the performance bound β is close to 1, it implies that the performance of the greedy solution is close to that of the global optimal solution (i.e., $f(Y^G) \simeq f(Y^*)$). Hence, β can also be seen as a measure of effectiveness of using a greedy algorithm to solve the problem (2).

For the considered class of submodular maximization problems in (2), the seminal paper (Nemhauser et al., 1978) has established the performance bound (denoted by β_f and referred to as the *fundamental performance bound*):

$$\beta_f \triangleq 1 - \left(1 - \frac{1}{N}\right)^N \leq \frac{f(Y^G)}{f(Y^*)}. \tag{4}$$

Note that $\beta_f > 1 - \frac{1}{e} \simeq 0.6321$ for any $N < \infty$ and $\lim_{N\to\infty} \beta_f = (1 - \frac{1}{e})$. This means that for any submodular maximization problem of the form (2), the corresponding greedy solution will always perform not worse than 63.21% of the maximum achievable performance level.

Remark 1. Upon obtaining a greedy solution, there are numerous established ways to improve it further. For example, Sun et al. (2019, 2020) proposed (continuous) gradient ascent processes while Nemhauser et al. (1978) and Welikala and Cassandras (2020, 2021) proposed (discrete) interchange schemes. Even though this paper does not consider such an improvement scheme, the following proposition can be made regarding how an established performance bound for a greedy solution should be modified upon an improvement to that greedy solution.

Proposition 1. *Consider a greedy solution Y^G that has a performance bound β. Upon executing an improvement scheme on Y^G, if $\bar{Y}^G \in \mathscr{I}^N$ is found with $f(\bar{Y}^G) > f(Y^G)$, its performance bound is $\bar{\beta} \triangleq \beta * \frac{f(\bar{Y}^G)}{f(Y^G)} \leq \frac{f(\bar{Y}^G)}{f(Y^*)}$ and $\bar{\beta} > \beta$.*

Proof. The proof follows from multiplying both sides of (3) by $\frac{f(\bar{Y}^G)}{f(Y^G)}$ and noticing the fact that $f(\bar{Y}^G) > f(Y^G)$.

Finally, note that in the sequel we use the notation $Z^i = \{z^1, z^2, \ldots, z^i\}$ (with $Z^0 = \emptyset$ and $Z^N = Y^G$) to represent the greedy solution constructed after running i greedy iterations in Algorithm 1. Note also that even though Algorithm 1 is limited to running only N greedy iterations, we use this (Z^i, z^i) notation more liberally for any $i \in \{0, 1, 2, \ldots, M\}$.

3. A brief review of existing performance bounds

In this section, we briefly review several tighter performance bounds (i.e., closer to 1 compared to β_f in (4)) established in the literature for the greedy solution given by Algorithm 1 for the class of problems in (2). To the best of the authors' knowledge, the list of performance bounds reviewed here is exhaustive. Note also that even though we consider the same greedy solution Y^G, having a tighter performance bound is still important as it allows us to: (i) have a more accurate sense of proximity of Y^G to the global optimal Y^* and (ii) make more informed decisions regarding spending extra resources to seek an improved solution (as mentioned in Remark 1).

As we will see next, each of these tighter performance bounds has been established using a *curvature* measure that characterizes the structural properties of the objective function f, the ground set X and the feasible space $\mathscr{I}^N$ involved in the considered problem (2). In particular, four such established curvature measures and their respective performance bounds are briefly reviewed in this section, outlining their properties, strengths and weaknesses.

3.1. *Total curvature (Conforti & Cornuéjols, 1984)*

For the problem (2), the *total curvature* measure α_t is defined as

$$\alpha_t \triangleq \max_{x \in X} \left[1 - \frac{\Delta f(x|X\backslash\{x\})}{\Delta f(x|\emptyset)}\right]. \tag{5}$$

The corresponding performance bound β_t is given by

$$\beta_t \triangleq \frac{1}{\alpha_t}\left[1 - \left(1 - \frac{\alpha_t}{N}\right)^N\right] \leq \frac{f(Y^G)}{f(Y^*)}. \tag{6}$$

Note that $0 \leq \alpha_t \leq 1$ and β_t is a decreasing function with respect to α_t. Therefore, if $\alpha_t \to 0$, the corresponding performance bound $\beta_t \to 1$. In contrast, if $\alpha_t \to 1$, the corresponding performance bound $\beta_t \to \beta_f$ (recall that β_f is given in (4)).

Note that the α_t expression in (5) can be written as

$$\alpha_t = 1 - \min_{x \in X} \left[\frac{\Delta f(x|X\backslash\{x\})}{\Delta f(x|\emptyset)}\right].$$

Since f is submodular over X, $\Delta f(x|X\backslash\{x\}) \leq f(x|\emptyset), \forall x \in X$. Therefore, α_t will provide an improved performance bound (i.e., $\beta_t \to 1$) if f and X in (2) are such that $\Delta f(x|\emptyset) \simeq \Delta f(x|X\backslash\{x\}), \forall x \in X$, i.e., in other words, if the submodularity property is *weak* (see also Remark 2 given below). Moreover, as $\Delta f(x|X\backslash\{x\}) = f(X) - f(X\backslash\{x\})$, evaluating α_t requires evaluating $f(X)$ - which in some applications might be ill-defined (e.g., see Sun et al., 2020) and also computationally expensive (as often $f(Y)$ is of complexity $O(|Y|)$).

Remark 2. In the remainder of this paper, we use notions of "weak" and "strong" to qualify the submodularity; they can be inferred based on the respective qualitative properties: the weakness or strength of satisfaction of an inequality based on the submodularity property. For example, as discussed above, we refer to the submodularity property of f over X as weak or strong based on whether $\Delta f(x|X\backslash\{x\}) \simeq f(x|\emptyset), \forall x \in X$ or $\Delta f(x|X\backslash\{x\}) \ll f(x|\emptyset), \forall x \in X$, respectively.

3.2. Greedy curvature (Conforti & Cornuéjols, 1984)

The *greedy curvature* measure α_g is computed based on successive greedy solutions that the greedy algorithm generates (i.e., based on $Z^0, \dots, Z^N$). Specifically, α_g is defined as

$$\alpha_g \triangleq \max_{0\leq i\leq N-1}\left[\max_{x\in X^i}\left(1-\frac{\Delta f(x|Z^i)}{\Delta f(x|\emptyset)}\right)\right], \tag{7}$$

where $X^i \triangleq \{x : x \in X\backslash Z^i, (Z^i \cup \{x\}) \in \mathscr{I}^N\}$ (the set of feasible options in the $(i+1)$th greedy iteration). The corresponding performance bound β_g is given by

$$\beta_g \triangleq 1-\alpha_g\left(1-\frac{1}{N}\right) \leq \frac{f(Y^G)}{f(Y^*)}. \tag{8}$$

Note that $0 \leq \alpha_g \leq 1$ and β_g is a decreasing function in α_g. Therefore, when $\alpha_g \to 0$, $\beta_g \to 1$. However, when $\alpha_g \to 1$, unlike in the case of β_t, $\beta_g \to \frac{1}{N} < \beta_f$.

The expression of α_g in (7) can be written as

$$\alpha_g = 1-\min_{0\leq i\leq N-1}\left[\min_{x\in X^i}\left(\frac{\Delta f(x|Z^i)}{\Delta f(x|\emptyset)}\right)\right].$$

Since f is submodular, $\Delta f(x|Z^i) \leq \Delta f(x|\emptyset)$. Therefore, to get an improved performance bound (i.e., $\beta_g \to 1$), f, X and $Z^i, i \in \{0, 1, \dots, N\}$ of the problem (2) should be such that $\Delta f(x|Z^i) \simeq \Delta f(x|\emptyset), \forall x \in X\backslash Z^i, i \in \{0, 1, \dots, N-1\}$, i.e., in other words, the submodularity property should be *weak*. Moreover, note that β_g in (8) can be computed in parallel with the execution of the greedy algorithm without requiring any additional numerical evaluations of f. Hence, unlike β_t in (6), β_g is computationally inexpensive as well as always fully-defined.

3.3. Elemental curvature (Wang et al., 2016)

For the problem (2), the *elemental curvature* measure α_e is defined as

$$\alpha_e \triangleq \max_{\substack{(Y,x_i,x_j):Y\subset X,\\ x_i,x_j\in X\backslash Y,\ x_i\neq x_j.}}\left[\frac{\Delta f(x_i|Y\cup\{x_j\})}{\Delta f(x_i|Y)}\right]. \tag{9}$$

The corresponding performance bound β_e is given by

$$\beta_e \triangleq 1-\left(\frac{\alpha_e+\alpha_e^2+\cdots+\alpha_e^{N-1}}{1+\alpha_e+\alpha_e^2+\cdots+\alpha_e^{N-1}}\right)^N \leq \frac{f(Y^G)}{f(Y^*)}. \tag{10}$$

Note that $0 \leq \alpha_e \leq 1$ and β_e is a decreasing function with respect to α_e. Therefore, when $\alpha_e \to 0$, $\beta_e \to 1$ and when $\alpha_e \to 1$, similar to the case of β_t, $\beta_e \to \beta_f$.

It can be shown that f is submodular over X if and only if $\Delta f(x_i|Y\cup\{x_j\}) \leq \Delta f(x_i|Y)$ for all feasible (Y, x_i, x_j) choices considered in (9) (see Nemhauser et al., 1978, Prop. 2.1). Therefore, if $\Delta f(x_i|Y\cup\{x_j\}) = \Delta f(x_i|Y)$ occurs for some feasible combination of (Y, x_i, x_j), it means f is *modular* in that region. According to (9), such an existence of a modular region of f over X causes $\alpha_e = 1$ resulting $\beta_e = \beta_f$. A trivial situation where this ($\beta_e = \beta_f$) occurs is when f and X in problem (2) are such that $\exists x_i, x_j \in X$ with $x_i \neq x_j$ where $f(\{x_i\}) + f(\{x_j\}) = f(\{x_i, x_j\})$. Therefore, it is clear that the elemental curvature based performance bound β_e fails (i.e., $\beta_e = \beta_f$ occurs) unless f is *strictly submodular* everywhere over its domain, i.e., $\Delta f(x_i|Y\cup\{x_j\}) \ll \Delta f(x_i|Y)$ for all feasible (Y, x_i, x_j) choices. We highlight that this particular behavior of β_e contrasts from that of β_t and β_g discussed before (where *weakly submodular* scenarios were preferred).

Moreover, note that evaluating β_e is significantly computationally expensive (even compared to β_t) as α_e in (9) involves solving a set function maximization problem (notice the set variable Y in (9)). Hence, such a problem can even be more complicated than the original set function maximization problem (2) that we consider unless there are some special structural properties that can be exploited (e.g., see Sun et al., 2019).

3.4. Partial curvature (Liu et al., 2018)

The motivation behind the *partial curvature* measure α_p is to be an alternative to the total curvature measure α_t in (5). Unlike α_t, α_p can be evaluated when f has a constrained domain, i.e., when $f : \mathscr{I} \to \mathbb{R}_{\geq 0}$ with $\mathscr{I} \subset 2^X$ (where α_t is ill-defined due to its $f(X)$ term). Specifically, α_p is defined as

$$\alpha_p = \max_{(Y,x):x\in Y\in\mathscr{I}^N}\left[1-\frac{\Delta f(x|Y\backslash\{x\})}{\Delta f(x|\emptyset)}\right]. \tag{11}$$

The corresponding performance bound β_p is given by

$$\beta_p \triangleq \frac{1}{\alpha_p}\left[1-\left(1-\frac{\alpha_p}{N}\right)^N\right] \leq \frac{f(Y^G)}{f(Y^*)}. \tag{12}$$

We highlight that the above β_p expression is only valid under a few additional conditions on f, X and $\mathscr{I}^N$ (which are omitted here, but can be found in Liu et al., 2018). Note that β_p in (12) and β_t in (6) has identical forms — enabling a direct comparison between α_t and α_p. The work in Liu et al. (2018) has shown that $\alpha_p \leq \alpha_t$, which implies that $\beta_p \geq \beta_t$, i.e., β_p is always tighter than β_t. Note also that, similar to β_t, β_p will provide a much improved performance bound (i.e., $\beta_p \to 1$) if the underlying submodularity property (of f over X) is *weak*.

Moreover, similar to α_e in (9), evaluating α_p in (11) involves solving a set function maximization problem (notice the set variable Y in (11)). Therefore, evaluating β_p is significantly computationally expensive compared to evaluating β_t. In fact, evaluating β_p can even be more complicated than the original set function maximization problem (2) that we consider unless there are some special structural properties that can be exploited (e.g., see Welikala, 2021, Ch. 3.2.4).

4. The new performance bound

From the review presented in the previous section, three main limitations of existing improved performance bounds (i.e., of β_t, Conforti & Cornuéjols, 1984, β_g, Conforti & Cornuéjols, 1984, β_e, Wang et al., 2016 and β_p, Liu et al., 2018) can be identified:

(1) **Computational complexity**: For example, β_e and β_p (i.e., the most recently proposed performance bounds) require solving hard combinatorial optimization problems.

(2) **Inherent limitations**: For example, β_t, β_g and β_p inherently provide improved performance bounds only when the submodularity property (of f over X) is weak.
(3) **Technical limitations**: For example, β_t and β_p have technical conditions that need to be satisfied (by f, X and $\mathscr{I}^N$ involved in (2)) to validate their usage.

To counter the limitations mentioned above, in this section, a new performance bound (denoted by β_u) is proposed for the greedy solution Y^G given by Algorithm 1 for the class of problems in (2). Similar to the previously reviewed improved performance bounds, this new performance bound β_u is also defined through a corresponding (also new) curvature measure. In particular, we denote this new curvature measure as α_u and call it the *extended greedy curvature*.

As the name suggests, this new curvature measure α_u (and hence β_u) is derived exploiting the information computed when executing an extended number of greedy iterations (i.e., more than the usual N greedy iterations executed in Algorithm 1). As we will see in the sequel, the exact number of extra greedy iterations required depends on the application and the user preference. Since running greedy iterations is computationally inexpensive, the complexity of computing β_u is much less than that of β_e or β_p and is in the same order of computing β_t or β_g. Moreover, as we will see in the sequel, unlike $\beta_t, \beta_g, \beta_e$ and β_p, β_u does not have any inherent or technical limitations.

4.1. Preliminary theoretical results

We start with establishing the following minor theoretical result that relates an *upper bound* found for the global optimal solution performance $f(Y^*)$ with a *performance bound* found for the greedy solution performance $f(Y^G)$.

Lemma 2. *Given the greedy solution performance $f(Y^G)$:*

(a) α is an upper bound for $f(Y^*)$ if and only if $\beta = \frac{f(Y^G)}{\alpha}$ is a valid performance bound for $f(Y^G)$.
(b) β is a valid performance bound for $f(Y^G)$ if and only if $\alpha = \frac{1}{\beta}f(Y^G)$ is an upper bound for $f(Y^*)$.

Proof. Cases (a) and (b) can be proved using the relationships: $\alpha \geq f(Y^*) \iff \frac{f(Y^G)}{\alpha} \leq \frac{f(Y^G)}{f(Y^*)}$ and $\beta \leq \frac{f(Y^G)}{f(Y^*)} \iff f(Y^*) \leq \frac{1}{\beta}f(Y^G)$, respectively.

We now introduce some additional notations. Let $[0, k]$ be the set $\{0, 1, 2, \ldots, k\}$. Recall the (Z^i, z^i) notation introduced in Section 2 for $i \in [0, M]$. Using that, let us define

$$Y_n^G \triangleq Z^{(n+1)N} \backslash Z^{nN} = \{z^{nN+1}, z^{nN+2}, \ldots, z^{nN+N}\}, \tag{13}$$

for any $n \in [0, m-1]$ where $m \triangleq \left\lfloor \frac{M}{N} \right\rfloor$ ($\lfloor\cdot\rfloor$ denotes the floor operator). Simply, Y_n^G is the $(n+1)$th block of N greedily selected options. Hence, $|Y_n^G| = N$ and $Y_0^G = Y^G$. Along the same lines, let us also define

$$X_n \triangleq X \backslash Z^{nN} \quad \text{and} \quad \mathscr{I}_n^N \triangleq \{Y : Y \subseteq X_n, |Y| \leq N\}, \tag{14}$$

for any $n \in [0, m-1]$. Simply, X_n is the set of remaining available options after selecting n blocks of N greedy options (i.e., after nN greedy iterations). Hence, $X_0 = X$ and $\mathscr{I}_0^N = \mathscr{I}^N$. Similar to the set system $(X, \mathscr{I}^N)$ considered in (2), the set system $(X_n, \mathscr{I}_n^N)$ is a uniform matroid of rank N, for any $n \in [0, m-1]$.

Let us also consider a series of auxiliary set function maximization problems: $\{\mathbf{P}_n\}_{n \in [0, m-1]}$ where

$$\mathbf{P}_n: \qquad Y_n^* \triangleq \underset{Y \in \mathscr{I}_n^N}{\arg\max}\ \Delta f(Y|Z^{nN}). \tag{15}$$

According to Lemma 1, the objective function of $\mathbf{P}_n$ (i.e., $\Delta f(Y|Z^{nN})$) is a polymatroid set function over X_n. This implies that $\mathbf{P}_n$ aims to find the optimal set Y_n^* that maximizes the polymatroid set function $\Delta f(Y|Z^{nN})$ over the uniform matroid $(X_n, \mathscr{I}_n^N)$. Hence, each $\mathbf{P}_n, n \in [0, m-1]$ falls into the same class of problems as in (2), and in fact, $\mathbf{P}_0$ is equivalent to (2) (i.e., $Y_0^* = Y^*$). Moreover, it is easy to see that Y_n^G introduced in (13) is the greedy solution to $\mathbf{P}_n$ in (15) for $n \in [0, m-1]$.

Next, we establish two lemmas that provide two different upper bounds for the global optimal performance of $\mathbf{P}_n$ in (15).

Lemma 3. *For $n \in [0, m-1]$,*

$$\Delta f(Y_n^*|Z^{nN}) \leq \max_{Y \in \mathscr{I}_n^N} \left[\sum_{y \in Y} \Delta f(y|Z^{nN})\right]. \tag{16}$$

Proof. Due to the normalized and monotone nature of the set function f (see Definition 1(a)–(b)), we have $0 \leq f(A \cap B)$ for all $A, B \subseteq X$. Using this result in the second equivalent condition given for the submodularity property in Definition 1(c), we can write $f(A \cup B) \leq f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$, i.e., $f(A \cup B) \leq f(A) + f(B)$ for all $A, B \subseteq X$ (notice the resemblance with the triangle inequality). Based on this result, it is easy to see that any normalized monotone submodular set function f defined over a ground set X will follow the property:

$$f(A) \leq \sum_{a \in A} f(\{a\}), \quad \forall A \subseteq X. \tag{17}$$

As mentioned before, $\Delta f(Y|Z^{nN})$ is a normalized monotone submodular set function in Y over the ground set X_n (from Lemma 1). Therefore, $\Delta f(Y|Z^{nN})$ should follow the property in (17), i.e.,

$$\Delta f(Y|Z^{nN}) \leq \sum_{y \in Y} \Delta f(y|Z^{nN}), \quad \forall Y \in \mathscr{I}_n^N, \tag{18}$$

(note that, according to (14), $Y \in \mathscr{I}_n^N \iff Y \subseteq X^n$). Now, taking the maximum of both sides of (18) over all possible $Y \in \mathscr{I}_n^N$, we get

$$\max_{Y \in \mathscr{I}_n^N} \Delta f(Y|Z^{nN}) \leq \max_{Y \in \mathscr{I}_n^N} \left[\sum_{y \in Y} \Delta f(y|Z^{nN})\right]. \tag{19}$$

Finally, using (15), we can rewrite the left hand side (LHS) of the above expression as $\Delta f(Y_n^*|Z^{nN})$. This completes the proof.

Lemma 4. *For $n \in [0, m-1]$,*

$$\Delta f(Y_n^*|Z^{nN}) \leq \frac{1}{\beta_f}\left[f(Z^{(n+1)N}) - f(Z^{nN})\right]. \tag{20}$$

Proof. Using (15), let us rewrite the LHS of (20) as

$$\Delta f(Y_n^*|Z^{nN}) = \max_{Y \in \mathscr{I}_n^N} \Delta f(Y|Z^{nN}). \tag{21}$$

Since $\Delta f(Y|Z^{nN})$ is a polymatroid set function in Y over the uniform matroid $(X_n, \mathscr{I}_n^N)$, the performance bound β_f given in (4) can be applied for a greedy solution of the above set function maximization problem (on right hand side of (21)). In fact, from the used notation, $Y = Y_n^G$ is the greedy solution that maximizes $\Delta f(Y|Z^{nN})$. Therefore, using the known performance bound β_f and the greedy solution performance $\Delta f(Y_n^G|Z^{nN})$ in Lemma 2(b), we obtain an upper bound to (21) as

$$\Delta f(Y_n^*|Z^{nN}) = \max_{Y \in \mathscr{I}_n^N} \Delta f(Y|Z^{nN}) \leq \frac{1}{\beta_f}\Delta f(Y_n^G|Z^{nN}). \tag{22}$$

Finally, we use (1) and (13) to simplify $\Delta f(Y_n^G|Z^{nN})$ as $\Delta f(Y_n^G|Z^{nN}) = f(Y_n^G \cup Z^{nN}) - f(Z^{nN}) = f(Z^{(n+1)N}) - f(Z^{nN})$. Substituting this result in (22) completes the proof.

The following lemma establishes an important equality condition that will be used later on.

Lemma 5. *For $n \in [0, m-1]$,*

$$\max_{Y \in \mathscr{I}_n^N} \Delta f(Y|Z^{(n+1)N}) = \max_{Y \in \mathscr{I}_{n+1}^N} \Delta f(Y|Z^{(n+1)N}). \tag{23}$$

Proof. Note that this result is non-trivial as the feasible spaces of the optimization problems on both sides of (23) are related such that $\mathscr{I}_n^N \supset \mathscr{I}_{n+1}^N$. Let us denote $Y = \{y_1, y_2, \dots, y_N\} \in \mathscr{I}_n^N$ and rewrite $\Delta f(Y|Z^{(n+1)N})$ as a telescoping sum:

$$\begin{aligned}\Delta f(Y|Z^{(n+1)N}) =& f(\{y_1, \dots, y_N\} \cup Z^{(n+1)N}) - f(Z^{(n+1)N})\\ =& f(\{y_1, \dots, y_N\} \cup Z^{(n+1)N})\\ & - f(\{y_1, \dots, y_{N-1}\} \cup Z^{(n+1)N}) + \cdots\\ & \cdots + f(\{y_1\} \cup Z^{(n+1)N}) - f(Z^{(n+1)N})\\ =& \sum_{i=1}^{N} \Delta f(y_i|\{y_1, \dots, y_{i-1}\} \cup Z^{(n+1)N}).\end{aligned} \tag{24}$$

Note that $\Delta f(y_i|\{y_1, \dots, y_{i-1}\} \cup Z^{(n+1)N}) = 0$ for any $y_i \in Z^{(n+1)N}$ and $\Delta f(y_i|\{y_1, \dots, y_{i-1}\} \cup Z^{(n+1)N}) > 0$ for any $y_i \in X \backslash Z^{(n+1)N}$. Therefore, according (24), when maximizing the set function $\Delta f(Y|Z^{(n+1)N})$ with respect to Y, selecting $Y \subset X \backslash Z^{(n+1)N}$ (a.k.a. $Y \in \mathscr{I}_{n+1}^N$) is sufficient as opposed to selecting $Y \subseteq X \backslash Z^{nN}$ (a.k.a. $Y \in \mathscr{I}_n^N$). Hence (23) holds.

Now, we establish a lemma that provides an upper bound for the performance of the global optimal solution of (2).

Lemma 6. *For $n \in [0, m-1]$,*

$$f(Y^*) \leq f(Z^{nN}) + \Delta f(Y_n^*|Z^{nN}). \tag{25}$$

Proof. Since $\Delta f(Y|Z^{nN})$ is a monotone set function in Y over the ground set X_n (from Lemma 1, for any $n \in [0, m-1]$),

$$\begin{aligned}\Delta f(Y_n^*|Z^{nN}) \leq& \Delta f(Y_n^* \cup Y_n^G|Z^{nN}),\\ =& \Delta f(Y_n^G|Z^{nN}) + \Delta f(Y_n^* \cup Y_n^G|Z^{nN}) - \Delta f(Y_n^G|Z^{nN})\\ =& \Delta f(Y_n^G|Z^{nN}) + f(Y_n^* \cup Y_n^G \cup Z^{nN}) - f(Z^{nN})\\ & - f(Y_n^G \cup Z^{nN}) + f(Z^{nN}) && \text{(using (1))}\\ =& \Delta f(Y_n^G|Z^{nN}) + f(Y_n^* \cup Y_n^G \cup Z^{nN}) - f(Y_n^G \cup Z^{nN})\\ =& \Delta f(Y_n^G|Z^{nN}) + \Delta f(Y_n^*|Y_n^G \cup Z^{nN}) && \text{(using (1))}\\ =& \Delta f(Y_n^G|Z^{nN}) + \Delta f(Y_n^*|Z^{(n+1)N}), && \text{(using (13))}\end{aligned}$$

i.e.,

$$\Delta f(Y_n^*|Z^{nN}) \leq \Delta f(Y_n^G|Z^{nN}) + \Delta f(Y_n^*|Z^{(n+1)N}). \tag{26}$$

Note that $\Delta f(Y_n^*|Z^{(n+1)N}) \leq \max_{Y \in \mathscr{I}_n^N} \Delta f(Y|Z^{(n+1)N})$ as $Y_n^* \in \mathscr{I}_n^N$. Using this result in (26), we can write

$$\begin{aligned}\Delta f(Y_n^*|Z^{nN}) \leq& \Delta f(Y_n^G|Z^{nN}) + \max_{Y \in \mathscr{I}_n^N} \Delta f(Y|Z^{(n+1)N})\\ =& \Delta f(Y_n^G|Z^{nN}) + \max_{Y \in \mathscr{I}_{n+1}^N} \Delta f(Y|Z^{(n+1)N}) && \text{(from Lemma 5)}\\ =& \Delta f(Y_n^G|Z^{nN}) + \Delta f(Y_{n+1}^*|Z^{(n+1)N}). && \text{(using (15))}\end{aligned}$$

Replacing n with k, the above result can be written as

$$\Delta f(Y_k^*|Z^{kN}) \leq \Delta f(Y_k^G|Z^{kN}) + \Delta f(Y_{k+1}^*|Z^{(k+1)N}), \tag{27}$$

for $k \in [0, m-1]$. Now summing (27) for $k \in [0, n-1]$ we get

$$\Delta f(Y_0^*|Z^0) \leq \sum_{k=0}^{n-1} \Delta f(Y_k^G|Z^{kN}) + \Delta f(Y_n^*|Z^{nN}). \tag{28}$$

Note that $Y_0^* = Y^*$ in (2) and $Z^0 = \emptyset$ by definition. Thus, $\Delta f(Y_0^*|Z^0) = f(Y^*)$. Further, using (1) and (13), we can show that

$$\begin{aligned}\sum_{k=0}^{n-1} \Delta f(Y_k^G|Z^{kN}) =& \sum_{k=0}^{n-1} \Delta f(Z^{(k+1)N} \backslash Z^{kN}|Z^{kN})\\ =& \sum_{k=0}^{n-1} f(Z^{(k+1)N}) - f(Z^{kN})\\ =& f(Z^{nN}).\end{aligned}$$

Therefore, using the above two results in (28), we now can obtain (25), which holds for any $n \in [0, m-1]$ (equality holds when $n = 0$).

4.2. Extended greedy curvature based performance bound

We now define the proposed extended greedy curvature measure α_u as

$$\alpha_u \triangleq \min_{i \in Q} \alpha_u^i, \tag{29}$$

where $Q \subseteq \bar{Q} \triangleq \{1, N, N+1, 2N, 2N+1, \dots, (m-1)N+1, mN, M\}$ and

$$\alpha_u^i \triangleq \begin{cases} f(Z^{i-1}) + \max_{Y \in \mathscr{I}_{(i-1)/N}^N} \left[\sum_{y \in Y} \Delta f(y|Z^{i-1})\right] & \text{if } i = 1, N+1, 2N+1, \dots, (m-1)N+1,\\ f(Z^{i-N}) + \frac{1}{\beta_f}\left[f(Z^i) - f(Z^{i-N})\right] & \text{if } i = N, 2N, \dots, mN,\\ f(Z^i) & \text{if } i = M. \end{cases} \tag{30}$$

We point out that $\bar{Q}$ is a fixed set of greedy iteration indexes upon each of which a corresponding α_u^i value can be computed using already known information. This is because all the $f(\cdot)$ and $\Delta f(\cdot|\cdot)$ terms required to evaluate any α_u^i form in (30) are automatically computed during the execution of first i greedy iterations. Hence, α_u^i sequence over $i \in \bar{Q}$ can be thought of as a sequence of byproducts generated during (in parallel with) the execution of greedy iterations. In contrast to $\bar{Q}$, Q is an arbitrary subset selected from $\bar{Q}$ based on the user preference. For example, one can simply set $Q = \{1, N, N+1, 2N\}$ so that α_u value can be obtained upon executing only N extra greedy iterations (i.e., $2N$ greedy iterations in total).

The performance bound β_u corresponding to the extended greedy curvature measure α_u is given in the following theorem.

Theorem 1. *For the submodular maximization problem in (2), the greedy solution Y^G given by Algorithm 1 satisfies the performance bound β_u where*

$$\beta_u \triangleq \frac{f(Y^G)}{\alpha_u} \leq \frac{f(Y^G)}{f(Y^*)}. \tag{31}$$

Proof. To prove this result, according to Lemma 2(a) and (29), we only need to show that $f(Y^*) \leq \alpha_u^i$ for all $i \in \bar{Q}$ (note also that $Q \subseteq \bar{Q}$). We do this in three steps.

First, by adding the main results established in Lemmas 6 and 3 (i.e., (25) and (16), respectively) we get

$$f(Y^*) \leq f(Z^{nN}) + \max_{Y \in \mathscr{I}_n^N} \left[\sum_{y \in Y} \Delta f(y|Z^{nN})\right], \tag{32}$$

for $n \in [0, m-1]$. Now, replacing the variable n with i using the substitution $i = nN + 1$ we obtain $f(Y^*) \leq \alpha_u^i$ for $i \in \{1, N+1, 2N+1, \dots, (m-1)N+1\}$.

Second, using the main results of Lemmas 4 and 6, we get

$$f(Y^*) \leq f(Z^{nN}) + \frac{1}{\beta_f}\left[f(Z^{(n+1)N}) - f(Z^{nN})\right], \tag{33}$$

for $n \in [0, m-1]$. Now, replacing the variable n with i using the substitution $i = nN + N$ we obtain $f(Y^*) \leq \alpha_u^i$ for $i \in \{N, 2N, 3N, \dots, mN\}$.

Finally, we use the monotonicity property of f and the fact that $Y^* \subseteq Z^M = X$ to obtain $f(Y^*) \leq f(Z^i) = \alpha_u^i$ for $i = M$.

4.3. Discussion

As mentioned earlier, the set Q used for computing α_u in (29) can be smaller compared to the set $\bar{Q}$ (specially if the computational cost associated with running extra greedy iterations is a concern). However, based on (29) and (31), it is easy to show that the performance bound β_u is a *monotone set function* in $Q \subseteq \bar{Q}$. This implies that a superset of Q will always provide a higher (or at least equal) β_u value compared to the β_u value corresponding to the set Q.

Let us now consider a case where the submodularity property of f is weak (i.e., f is close to being a modular set function, see also Remark 2). In a such setting, the α_u^1 value (from (30), with $i = 1$) can be simplified as

$$\begin{aligned}\alpha_u^1 &= f(Z^0) + \max_{Y \in \mathscr{I}_0^N}\left[\sum_{y\in Y}\Delta f(y|Z^0)\right] = \max_{Y\in\mathscr{I}^N}\left[\sum_{y\in Y} f(\{y\})\right] \\ &= \max_{Y\in\mathscr{I}^N} f(Y) + \epsilon = f(Y^*) + \epsilon,\end{aligned} \tag{34}$$

where $\epsilon \geq 0$ is a parameter that represents the strength of the submodularity property ($\epsilon = 0$ if f is modular). The above expression implies that the evaluated α_u^1 value will be a tight upper bound for $f(Y^*)$ as f become more modular (i.e., as $\epsilon \to 0$). Therefore, in a such setting, the corresponding performance bound β_u will also be tight (close to 1). In that sense, β_u behaves similar to the performance bounds $\beta_t, \beta_g, \beta_p$ discussed in the previous section.

Remark 3. The strength of the submodularity can formally be defined as an additive constant $\epsilon_a \in \mathbb{R}_{\geq 0}$ or a multiplicative constant $\epsilon_m \in \mathbb{R}_{\geq 0}$ applicable for the submodularity inequality provided in Definition 1(c). Based on this definition, it can be shown that the parameter ϵ we introduced in (34) (to represent the strength of the submodularity) satisfies $\epsilon \geq \epsilon_a$ and $\frac{\epsilon}{f(Y^*)} \geq \epsilon_m$.

On the other hand, let us now consider a case where the submodularity property of f is strong. In a such setting, according to the "diminishing returns" view of the submodularity property (see Definition 1(c)), $f(Z^i)$ should saturate quickly with respect to i. Keeping this in mind, let us consider the α_u^{2N} value (from (30), with $i = 2N$) that can be simplified as

$$\begin{aligned}\alpha_u^{2N} &= f(Z^N) + \frac{1}{\beta_f}\left[f(Z^{2N}) - f(Z^N)\right] \\ &= f(Y^G) + \frac{1}{\beta_f}\left[f(Z^{2N}) - f(Z^N)\right].\end{aligned}$$

Note that when the submodularity property of f is strong, the above difference term $\left[f(Z^{2N}) - f(Z^N)\right]$ will become small. Therefore, $\alpha_u^{2N} \to f(Y^G)$ and the corresponding performance bound $\beta_u = \frac{f(Y^G)}{\alpha_u^{2N}} \to 1$ revealing a tight performance bound. In that sense, β_u behaves similar to the performance bound β_e discussed in the previous section.

Therefore, β_u is designed to have the best of both worlds while also being computationally inexpensive and having no additional technical limitations on its applicability. In the next section, we confirm these conclusions using numerical results generated from several different experiments.

5. Application to multi-agent coverage problems

In this section, we consider a widely studied class of multi-agent coverage problems (Sun et al., 2019; Welikala & Cassandras, 2020; Zhong & Cassandras, 2011) and show that problems in this class can be modeled as submodular maximization problems of the form (2). Therefore, we use the simple greedy algorithm (Algorithm 1) to solve these multi-agent coverage problems. Subsequently, we study the effectiveness of different performance bounds (discussed in previous sections) in characterizing such greedy solutions.

5.1. Multi-agent coverage problem formulation

The multi-agent coverage problem aims to find an optimal arrangement for a given set of sensors (agents) inside a given mission space to maximize the probability of detecting randomly occurring events in that mission space.

The *mission space* $\Omega \subset \mathbb{R}^2$ is modeled as a non-self-intersecting polygon (Zhong & Cassandras, 2011) that may contain a finite set of polygonal *obstacles* $\{\tilde{M}_1, \tilde{M}_2, \dots, \tilde{M}_h\}$ where $\tilde{M}_i \subset \Omega$ represents the interior of the ith obstacle. Hence, the agent deployment is constrained to the feasible space $F = \Omega\backslash(\bigcup_{i=1}^h \tilde{M}_i)$. The likelihood of random event occurrence over the mission space is modeled by the *event density function* $R : \Omega \to \mathbb{R}_{\geq 0}$. It is assumed that $R(x) = 0, \forall x \notin F$ and $\int_\Omega R(x)dx < 0$. In the case where no prior information about $R(x)$ is available, $R(x) = 1, \forall x \in F$ is used.

Inside the feasible space F, N homogeneous agents must be placed. We use $\mathbf{s} = [s_1, s_2, \dots, s_N] \in \mathbb{R}^{2\times N}$ to represent the selected agent locations (i.e., the control variable). Each agent is assumed to have a finite *sensing radius* $\delta \in \mathbb{R}$ beyond which it cannot detect any events. Further, obstacles are assumed to obstruct the sensing capability of the agents. In particular, the *visibility region* of an agent located at $s_i \in F$ is denoted by $V(s_i) = \{x : \|x - s_i\| \leq \delta, \forall q \in [0,1], (qx + (1-q)s_i) \in F\}$ (where $\|\cdot\|$ represents the Euclidean norm, see also Fig. 1). Moreover, a sensing function $p(x, s_i) = e^{-\lambda\|x-s_i\|}\cdot \mathbf{1}\{x \in V(s_i)\}$ is used to quantify the probability of an agent located at $s_i \in F$ detecting an event occurring at $x \in F$. In this $p(x, s_i)$ expression, the parameter λ is called the *sensing decay rate*.

Assuming independently detecting agents, the probability of detecting an event occurring at $x \in F$ by at least one agent (when in an agent placement $\mathbf{s}$) is given by $P(x, \mathbf{s}) = 1 - \prod_{i=1}^N[1 - p(x, s_i)]$. This is more commonly known as the *joint detection probability function*. Using the event density function and the joint detection probability function, the objective function of the multi-agent coverage problem can be written as $H(\mathbf{s}) = \int_\Omega R(x)P(x, \mathbf{s})dx$. Therefore, the multi-agent coverage problem can be stated as

$$\mathbf{s}^* = \underset{\mathbf{s}: s_i\in F, \forall i\in[1,N]}{\arg\max}\ H(\mathbf{s}). \tag{35}$$

5.2. Set function approach for multi-agent coverage problems

To model the multi-agent coverage problem in (35) as a set function maximization problem of the form (2), we use the following set of steps. First, the ground set $X = \{x_1, x_2, \dots, x_M\}$ is created by discretizing the continuous feasible space $F \subset \mathbb{R}^2$. Next, a set variable is defined as $S = \{s_1, s_2, \dots, s_N\}$ to represent

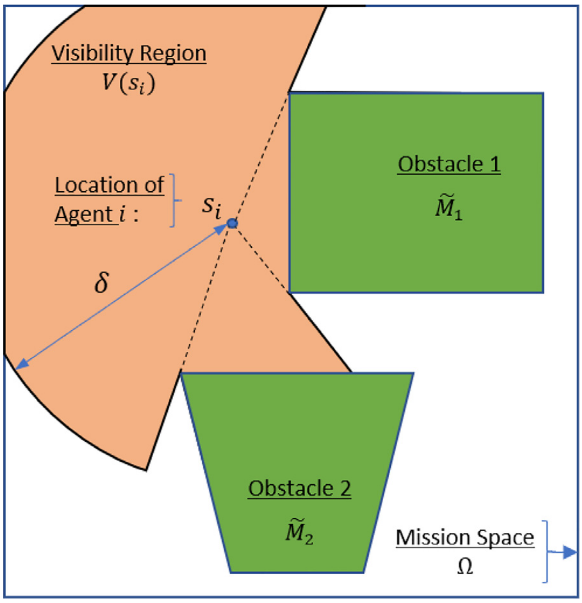


Fig. 1. A mission space with one agent.

the selected agent locations from the ground set X. Since we are interested in deploying only N agents, a uniform matroid constraint of rank N can be introduced as $S \in \mathscr{I}^N$ where $\mathscr{I}^N = \{Y : Y \subseteq X, |Y| \leq N\}$. Now the corresponding set function maximization problem for (35) can be written as

$$S^* = \underset{S \in \mathscr{I}^N}{\arg\max}\ H(S), \tag{36}$$

where $H(S)$ (called the *set coverage function*) is the set function version of the coverage objective function $H(\mathbf{s})$, i.e.,

$$H(S) = \int_F R(x)(1 - \prod_{s_i \in S}[1 - p(x, s_i)])dx. \tag{37}$$

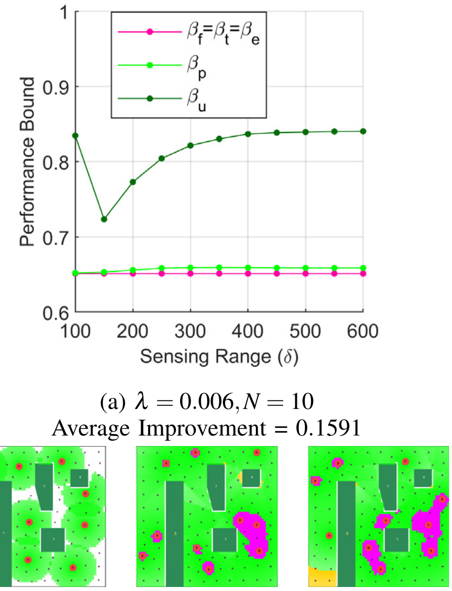
The work in Sun et al. (2019) has established that $H(S)$ in (37) is a polymatroid set function. Hence, it is clear that the multi-agent coverage problem in (35) is a submodular maximization problem of the form (2).

As a consequence, we now can solve the multi-agent coverage problem conveniently using the greedy algorithm (Algorithm 1) and also get a performance bound that characterizes how close the obtained greedy coverage level is to the global optimal coverage level. As mentioned in Remark 1, similar to the works in Sun et al. (2019, 2020), a subsequent gradient ascent stage or an interchange stage can be added to further improve any greedy solution obtained for (36). However, as shown in Proposition 1, such an improvement will only scale any performance bound (already found for the greedy solution) by a fixed constant factor. Therefore, in this paper, we omit executing such improvement stages and directly study the performance bounds found for the greedy solution.

5.3. Numerical results

The greedy algorithm (Algorithm 1), the newly proposed performance bound β_u (31) (α_u in (31) was determined using (29) with $Q = \bar{Q}$) and the existing other performance bounds: β_f (4), β_t (6), β_g (8), β_e (10) and β_p (12) were all implemented for the considered class of multi-agent coverage problems in an interactive JavaScript-based simulator which is available at http://www.bu.edu/codes/simulations/shiran27/CoverageFinal/ (the source code is available at: https://github.com/shiran27/CoverageControl). It may be used by the reader to reproduce the reported results and also to try different new problem configurations.

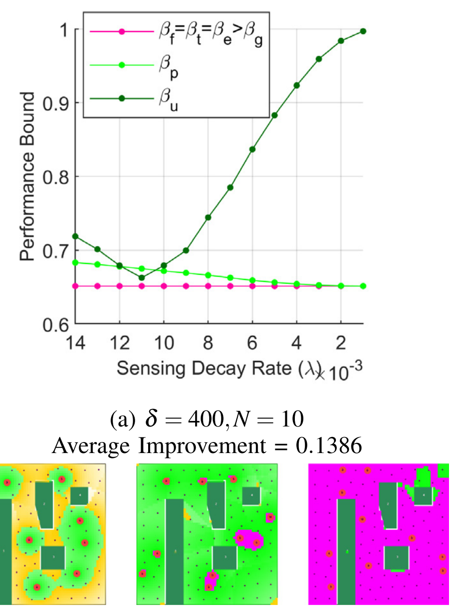
In particular, mission spaces with three different obstacle arrangements named 'General,' 'Maze' and 'Blank' were considered (can be seen in Figs. 2(b), 4(d) and 5(b), respectively). In



(a) $\lambda = 0.006, N = 10$
Average Improvement = 0.1591

(b) $\delta = 100$ (c) $\delta = 350$ (d) $\delta = 600$

Fig. 2. Performance bound vs. sensing range (δ) for the General mission space configuration. Sub-figures (b)–(d) show three greedy solutions obtained for three different δ values. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) $\delta = 400, N = 10$
Average Improvement = 0.1386

(b) $\lambda = 0.014$ (c) $\lambda = 0.007$ (d) $\lambda = 0.001$

Fig. 3. Performance bound vs. sensing decay rate (λ) for the General mission space configuration. Sub-figures (b)–(d) show three greedy solutions obtained for three different λ values. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

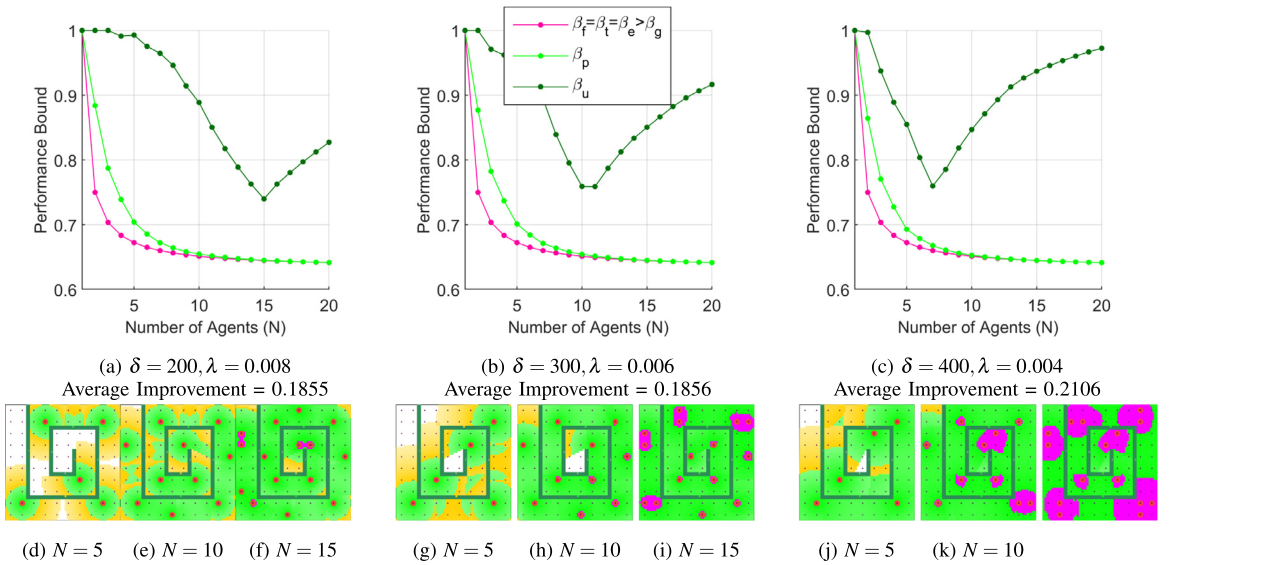


Fig. 4. Performance bound vs. number of agents (N) for the Maze mission space configuration: Agent sensing capabilities (or proportionately, the strength of the submodularity property of the set coverage objective) is: (a) weak, (b) moderate and (c) strong. Each of the corresponding three sub-figure groups (d)–(f), (g)–(i) and (j)–(l) shows three greedy solutions obtained for different N values. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

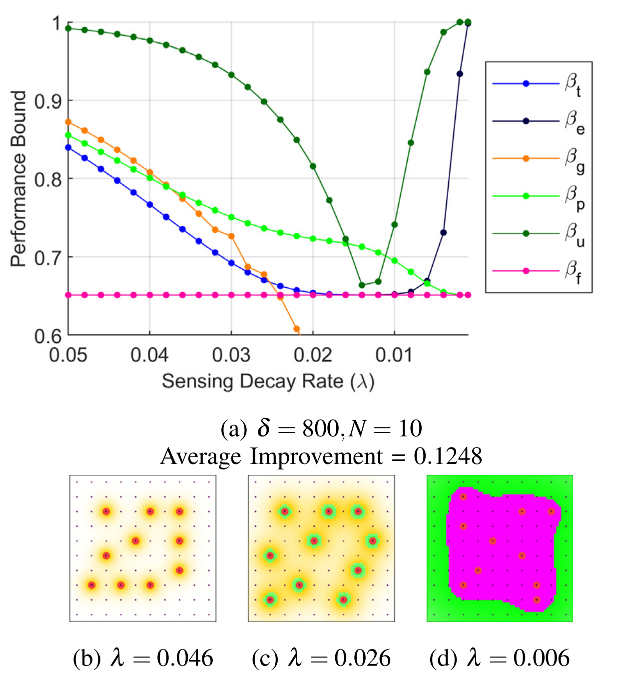


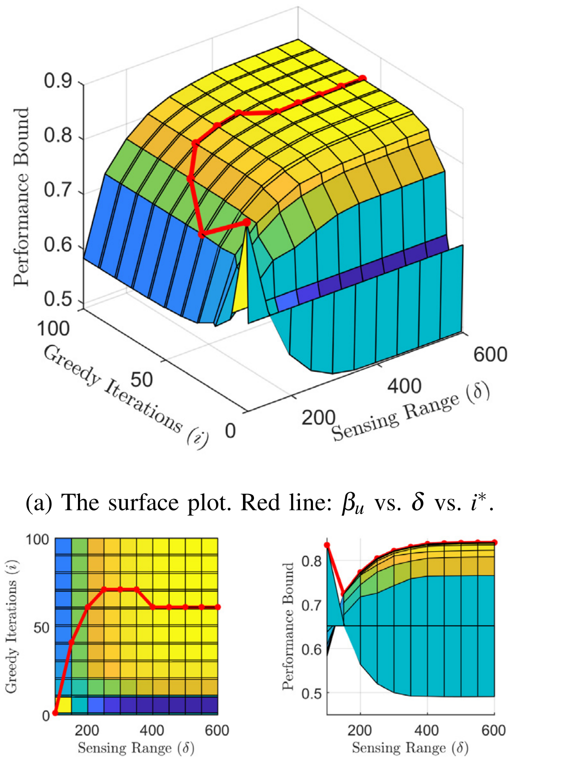
Fig. 5. Performance bound vs. sensing decay rate (λ) for the Blank mission space configuration. Sub-figures (b)–(d) show three greedy solutions obtained for three different λ values. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

such mission space diagrams, obstacles are shown as dark green-colored blocks and agent locations are shown as red-colored dots. Moreover, light-colored areas indicate low coverage levels while dark-colored areas indicate the opposite.

The main focus here is to study the behavior of different performance bounds (found for the greedy solution) under different coverage problem configurations. In each experiment shown in Figs. 3–5, one of the three parameters: (i) sensing range δ, (ii) sensing decay rate λ or (iii) the number of deployed agents N, was varied while keeping the other two fixed. Note that as δ increases or λ decreases or N increases, the *agent sensing capability* over the mission space also increases. For convenience, each graph has been drawn so that along its x-axis, the agent sensing capability increases. It is easy to see that agent sensing capability directly maps to the strength of the submodularity property of the set coverage objective function (i.e., agents with high sensing capabilities make the submodularity property of the corresponding set coverage objective function strong, and vice versa). In the graphs shown in Figs. 2(a), 3(a), 4(a), 4(b), 4(c) and 5(a), whenever only a few of the said performance bounds have been drawn, it means the other performance bounds were found redundant (no better than β_f). The *average improvement* value reported in each such graph (caption) was computed by taking the average of $(\beta_u - \max\{\beta_f, \beta_t, \beta_g, \beta_e, \beta_f\})$ value across all the corresponding data points.

Across almost all the numerical results shown (see Fig. 3(a), $\lambda = 11 \times 10^{-3}$ case for an exception), the proposed extended greedy curvature based performance bound β_u has shown the best performance bounds irrespective of the level of the agent sensing capabilities (i.e., irrespective of the strength of the submodularity property). In particular, the dip in the β_u curve seen in each of Figs. 2, 3, 4(a), 4(b), 4(c) and 5 point out that using a greedy algorithm to solve a multi-agent coverage problem is most challenging when the agents have a moderate sensing capability. Therefore, in such scenarios, the use of a greedy solution improvement scheme (see Remark 1) can be recommended. We also highlight that in each β_u curve, the decreasing set of data points (to the left of the dip point) have come from the extended greedy curvature measure $\alpha_u = \alpha_u^1$ (that can be computed in the very first greedy iteration). Hence, in all such scenarios, no extra greedy iterations were required.

To further study this, let us denote i^* as the argmin value of the problem (29). In other words, i^* is the minimum number of greedy iterations required to obtain the extended greedy curvature measure $\alpha_u = \alpha_u^{i^*}$ (29) and the corresponding performance bound $\beta_u = f(Y^G)/\alpha_u^{i^*}$ (31). For the same experiments that generated the results shown in Figs. 2(a) and 3(a), the corresponding

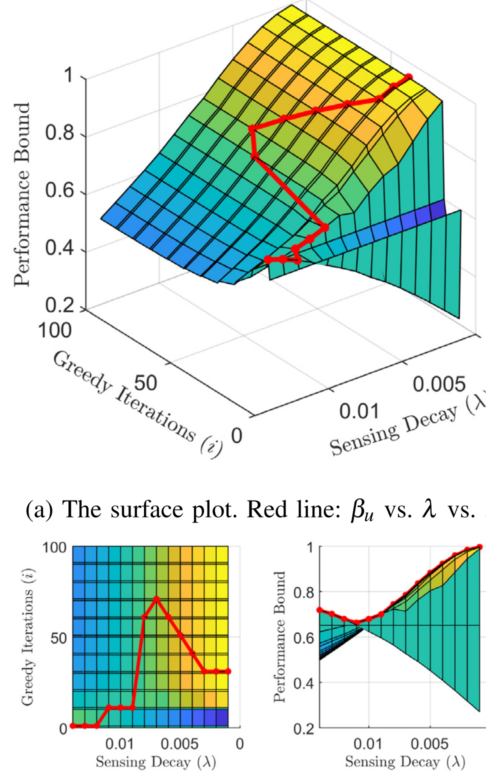


(a) The surface plot. Red line: β_u vs. δ vs. i^*.

(b) XY-View

(c) XZ-View (Fig. 2(a))

Fig. 6. Extend greedy curvature based performance bound vs. sensing range (δ) vs. no. of greedy iterations for the General mission space configuration with $\lambda = 0.006$, $N = 10$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) The surface plot. Red line: β_u vs. λ vs. i^*.

(b) XY-View

(c) XZ-View (Fig. 3(a))

Fig. 7. Extend greedy curvature based performance bound vs. sensing decay rate (λ) vs. no. of greedy iterations for the General mission space configuration with $\delta = 400$, $N = 10$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

i^* vs β_u behaviors observed (under different λ and α values) are illustrated in Figs. 6 and 7, respectively. Note that the red curves in Figs. 6(b) and 7(b) imply that the number of greedy iterations required to compute the proposed extended greedy curvature based performance bound β_u (31) (i.e., i^*) is generally non-trivial and typically $N < i^* < M$. Note also that the same curves indicate the need for a higher number of greedy iterations to compute the proposed performance bound β_u (31) (i.e., $i^* \gg N$) whenever the agents have a moderate sensing capability. This observation is in line with our previous conclusion that a multi-agent coverage problem is most challenging when the agents have a moderate sensing capability.

Comparison with the previous work in (Sun et al., 2019, 2020). For this class of multi-agent coverage problems, the work in Sun et al. (2019) first proposed to adopt the performance bounds β_t (6) and β_e (10) (from Conforti & Cornuéjols, 1984 and Wang et al., 2016, respectively). Then, the subsequent work in Sun et al. (2020) proposed to adopt the performance bounds β_g (8) and β_p (12) (from Conforti & Cornuéjols, 1984 and Liu et al., 2018, respectively). The numerical results shown in Fig. 5 justify these contributions of Sun et al. (2019, 2020) as they have lead to improved performance bounds compared to β_f. However, even in this case (Fig. 5), it is notable that the proposed novel performance bound in this paper β_u (31) has achieved an average improvement of 0.1248 compared to the state of the art (i.e., compared to $\max\{\beta_f, \beta_t, \beta_e, \beta_g, \beta_p\}$).

6. Conclusion

In this paper, we considered the class of monotone submodular set function maximization problems subject to cardinality constraints. Different curvature measures and corresponding performance bounds found in the literature were reviewed for this class of problems, outlining their strengths and weaknesses. In particular, computational complexity, technical requirements and inherent limitations were the main weaknesses observed. A novel curvature measure was proposed along with a corresponding performance bound that does not suffer from the limitations identified in its predecessors. We named this curvature measure as the *extended greedy curvature* since it thrives on the information seen when executing additional greedy iterations. A well-known class of multi-agent coverage problems was used to examine the effectiveness of the proposed performance bound compared to the other performance bounds found in the literature. Ongoing research explores the effectiveness of this new performance bound on other applications and under different quantified strength levels of the submodularity property.

Acknowledgments

We are immensely grateful to Prof. David A. Castañón, Prof. Sean B. Andersson and Prof. Ioannis Ch. Paschalidis of Boston University, Brookline, MA, USA, for sharing their expertise, insights and comments on an earlier version of this manuscript.

References

Badanidiyuru, Ashwinkumar, Karbasi, Amin, Kazemi, Ehsan, & Vondrak, Jan (2020). Submodular maximization through barrier functions. In *Advances in neural information processing systems*: *vol. 33*, (pp. 524–534).

Conforti, Michele, & Cornuéjols, Gérard (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.

Corneuejols, Gerard, Fisher, Marshall L., & Nemhauser, George L. (1977). Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, *23*(8), 789–810. http://dx.doi.org/10.1287/mnsc.23.8.789.

Das, Abhimanyu, & Kempe, David (2008). Algorithms for Subset Selection in Linear Regression. In *Proc. of 40th annual ACM symposium on theory of computing* (pp. 45–54).

El-Arini, Khalid, & Guestrin, Carlos (2011). Beyond Keyword Search: Discovering Relevant Scientific Literature. In *Proc. of 17th ACM SIGKDD intl. conf. on knowledge discovery and data mining* (pp. 439–447).

Fahrbach, Matthew, Mirrokni, Vahab, & Zadimoghaddam, Morteza (2019). Non-monotone submodular maximization with nearly optimal adaptivity and query complexity. In *Proc. of the 36th intl. conf. on machine learning*: *vol. 97*, (pp. 1833–1842).

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.

Grötschel, M., Lovász, L., & Schrijver, A. (1981). The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, *1*(2), 169–197. http://dx.doi.org/10.1007/BF02579273.

Jegelka, Stefanie, & Bilmes, Jeff (2011). Submodularity beyond submodular energies: Coupling edges in graph cuts. In *Proc. of IEEE computer society conf. on computer vision and pattern recognition* (pp. 1897–1904). http://dx.doi.org/10.1109/CVPR.2011.5995589.

Kempe, David, Kleinberg, Jon, & Tardos, Eva (2003). Maximizing the Spread of Influence through a Social Network. In *Proc. of 9th ACM SIGKDD intl. conf. on knowledge discovery and data mining* (pp. 137–146).

Khanna, Rajiv, Elenberg, Ethan, Dimakis, Alex, Negahban, Sahand, & Ghosh, Joydeep (2017). Scalable greedy feature selection via weak submodularity. In *Proc. of 20th intl. conf. on artificial intelligence and statistics*: *vol. 54*, (pp. 1560–1568).

Lin, Hui, & Bilmes, Jeff (2011). A class of submodular functions for document summarization. In *Proc. of 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 510–520).

Liu, Siwen (2020). A review for submodular optimization on machine scheduling problems. In *Complexity and approximation: In memory of Ker-I Ko* (pp. 252–267). http://dx.doi.org/10.1007/978-3-030-41672-0_16.

Liu, Yajing, Chong, Edwin K. P., & Pezeshki, Ali (2018). Improved bounds for the greedy strategy in optimization problems with curvature. *Journal of Combinatorial Optimization*, *37*(4), 1126–1149.

Liu, Yajing, Chong, Edwin K. P., Pezeshki, Ali, & Zhang, Zhenliang (2020). Submodular optimization problems and greedy strategies: A survey. *Discrete Event Dynamic Systems: Theory and Applications*, *30*(3), 381–412. http://dx.doi.org/10.1007/s10626-019-00308-7.

Mirzasoleiman, Baharan, Karbasi, Amin, Sarkar, Rik, & Krause, Andreas (2013). Distributed submodular maximization: Identifying representative elements in massive data. In *Advances in neural information processing systems*: *vol. 26*.

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.

Rezazadeh, Navid, & Kia, Solmaz S. (2019). A sub-modular receding horizon approach to persistent monitoring for a group of mobile agents over an urban area. *IFAC-PapersOnLine*, *52*(20), 217–222.

Schrijver, Alexander (2000). A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory. Series B*, *80*(2), 346–355. http://dx.doi.org/10.1006/jctb.2000.1989.

Sun, Xinmiao, Cassandras, Christos G., & Meng, Xiangyu (2019). Exploiting submodularity to quantify near-optimality in multi-agent coverage problems. *Automatica*, *100*, 349–359.

Sun, Chuangchuang, Welikala, Shirantha, & Cassandras, Christos G. (2020). Optimal composition of heterogeneous multi-agent teams for coverage problems with performance bound guarantees. *Automatica*, *117*, Article 108961. http://dx.doi.org/10.1016/j.automatica.2020.108961.

Wang, Zengfu, Moran, Bill, Wang, Xuezhi, & Pan, Quan (2016). Approximation for maximizing monotone non-decreasing set functions with a greedy method. *Journal of Combinatorial Optimization*, *31*(1), 29–43.

Welikala, Shirantha (2021). *Overcoming local optima in control and optimization of cooperative multi-agent systems* (Ph.D. thesis), (p. 233). Boston University.

Welikala, Shirantha, & Cassandras, Christos G. (2020). Asymptotic analysis for greedy initialization of threshold-based distributed optimization of persistent monitoring on graphs. *53*, In *Proc. of 21st IFAC world congress* (2), (pp. 3433–3438). http://dx.doi.org/10.1016/j.ifacol.2020.12.1670, URL https://www.sciencedirect.com/science/article/pii/S2405896320322734.

Welikala, Shirantha, & Cassandras, Christos G. (2020). Distributed non-convex optimization of multi-agent systems using boosting functions to escape local optima. *IEEE Transactions on Automatic Control*, http://dx.doi.org/10.1109/TAC.2020.3034869.

Welikala, Shirantha, & Cassandras, Christos G. (2021). Greedy initialization for distributed persistent monitoring in network systems. *Automatica*, *134*, 109943. http://dx.doi.org/10.1016/j.automatica.2021.109943.

Wolsey, Laurence A. (1982). Maximising real-valued submodular functions: Primal and dual heuristics for location problems. *Mathematics of Operations Research*, *7*(3), 410–425.

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.

Shirantha Welikala received the B.Sc. degree in Electrical and Electronic Engineering from the University of Peradeniya, Peradeniya, Sri Lanka, in 2015 and the M.Sc. and the Ph.D. degrees in Systems Engineering from Boston University, Brookline, MA, USA, in 2019 and 2021, respectively. From 2015 to 2017, he was with the Department of Electrical and Electronic Engineering, University of Peradeniya, where he worked first as a Temporary Instructor and subsequently as a Research Assistant. He is currently a Postdoctoral Research Fellow in the Department of Electrical Engineering, University of Notre Dame, South Bend, IN, USA. His main research interests include control and optimization of cooperative multi-agent systems with a particular emphasis on coverage and monitoring applications, networked systems, passivity, symbolic control, machine-learning, robotics, and smart-grid applications. He is a recipient of several awards, including the 2015 Ceylon Electricity Board Gold Medal, the 2019 President's Award for Scientific Research in Sri Lanka, and the 2021 Outstanding Ph.D. Dissertation Award in Systems Engineering.

Christos G. Cassandras (F'96) 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, 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 simulation, with applications to computer and sensor networks, manufacturing systems, and transportation systems. He has published about 450 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 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, 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.

Hai Lin is a professor at the Department of Electrical Engineering, University of Notre Dame, where he got his Ph.D. in 2005. Before returning to his *alma mater*, he worked as an assistant professor in the National University of Singapore from 2006 to 2011. Dr. Lin's teaching and research activities focus on the multidisciplinary study of fundamental problems at the intersections of control theory, machine learning and formal methods. His current research thrust is motivated by challenges in cyber–physical systems, long-term autonomy, multi-robot cooperative tasking, and human–machine collaboration. Dr. Lin has served on several committees and editorial boards, including *IEEE Transactions on Automatic Control*. He served as the chair for the IEEE CSS Technical Committee on Discrete Event Systems from 2016 to 2018, the program chair for IEEE ICCA 2011, IEEE CIS 2011 and the chair for IEEE Systems, Man and Cybernetics Singapore Chapter for 2009 and 2010. He is a senior member of IEEE and a recipient of 2013 NSF CAREER award.

Panos Antsaklis is the H.C. & E.A. Brosey Professor of Electrical Engineering at the University of Notre Dame. He is graduate of the National Technical University of Athens, Greece, and holds MS and Ph.D. degrees from Brown University. His research addresses problems of control and automation and examines ways to design control systems that will exhibit high degree of autonomy. His current research focuses on Cyber–Physical Systems and the interdisciplinary research area of control, computing and communication networks, and on hybrid and discrete event dynamical systems. He is IEEE, IFAC and AAAS Fellow, President of the Mediterranean Control Association, the 2006 recipient of the Engineering Alumni Medal of Brown University and holds an Honorary Doctorate from the University of Lorraine in France. He served as the President of the IEEE Control Systems Society in 1997 and was the Editor-in-Chief of the IEEE Transactions on Automatic Control for 8 years, 2010–2017.