Distributed Non-Convex Optimization of Multi-Agent Systems Using Boosting Functions to Escape Local Optima

Shirantha Welikala and Christos G. Cassandras

Abstract—**We address the problem of multiple local optima arising due to non-convex objective functions in cooperative multi-agent optimization problems. To escape such local optima, we propose a systematic approach based on the concept of *boosting functions*. The underlying idea is to temporarily transform the gradient at a local optimum into a *boosted gradient* with a non-zero magnitude. We develop a Distributed Boosting Scheme (DBS) based on a gradient-based optimization algorithm using a novel optimal variable step size mechanism so as to guarantee convergence. Even though our motivation is based on the coverage control problem setting, our analysis applies to a broad class of multi-agent problems. Simulation results are provided to compare the performance of different boosting functions families and to demonstrate the effectiveness of the boosting function approach in attaining improved (still generally local) optima.**

Index Terms—**Multi-Agent Systems, Distributed Optimization, Non-Convex Optimization, Boosting Functions.**

I. Introduction

A cooperative multi-agent system consists of interacting *agents* (subsystems) where each agent is allowed to control its local state under various constraints so as to collectively optimize a common global objective which depends on all the agent states. In this context, the goal of a *distributed* optimization scheme is to drive the multi-agent system to a globally optimal state by employing a controller at each agent which uses only the locally available information. Cooperative multi-agent systems can be classified depending on factors such as: (i) the nature of the agents (e.g., whether they are sensors, robots, vehicles, supply sources, processor cores, etc.), (ii) the constraints on their (local) decision space, (iii) the inter-agent interactions, and, (iv) the form of the global objective function. As a result, a large number of optimization methods are found in the literature which have focused on addressing different classes of multi-agent systems. Specifically, cooperative multi-agent systems play a major role in coverage control [1], formation control [2], learning [3], resource allocation [4], monitoring [5], consensus [6], transportation [7] and the smart grid [8].

While computationally complex optimization schemes are gaining interest due to their *generality*, gradient-based techniques are still widely popular due to their *simplicity* [9]. In cases where the objective function of the cooperative multi-agent system is convex, convergence to the global optimum can normally be guaranteed. The Alternating Direction Method of Multipliers (ADMM) [10] and Relaxed-ADMM method proposed in [11] are such examples. However, when the objective function is non-convex, a similar guarantee cannot be achieved [1], [12], [9], [10] without resorting to global optimization approaches. For example, simulated annealing [13], [14], particle swarm algorithms [15] and genetic algorithms [16] are some commonly used global optimization methods which are computationally intensive. The critical underlying common feature to these methods is the randomness introduced in controlling agents.

Inspired by this idea, the ladybug exploration method proposed in [17] suggests to hover over probable local optima aiming to find a better optimum. Similarly, randomness is introduced (or exploited) in the distributed non-convex consensus problems studied in [18] and [19] (which respectively use stochastic gradient and perturbed push-sum based algorithms) to converge to a local optimum while assuming the absence of saddle points. Efficiently escaping saddle points is by itself a challenging problem which again can be solved by introducing randomness. For example, [20] and [21] respectively study a perturbed gradient descent algorithm that can evade saddle points under a centralized setting. However, as pointed out before, although such optimization approaches that exploit some introduced randomness are more generally applicable, due to their computational complexity, they are infeasible for many applications.

Due to these reasons, addressing the issue of non-convexity without compromising simplicity has recently attracted renewed attention. In such methods, properties of the objective function are usually exploited. For example, the work in [22], [23] has proposed a combined greedy-gradient approach which utilizes the submodularity property of many non-convex objective functions. Balanced detection [1] and local optima smoothing [24] are two other approaches where the structure of the objective function has been used to trade-off between local approximation and global exploration in achieving better local optima. Along the same lines, [12] introduces the concept of "boosting functions" which can provide an alternative set of "boosted" gradients for agents to follow so that they can escape any local optimum (including saddle points, see also

[*] Supported in part by NSF under grants ECCS-1931600, DMS-1664644, CNS-1645681, by AFOSR under grant FA9550-19-1-0158, by ARPA-E's NEXTCAR program under grant DE-AR0000796 and by the MathWorks.

Authors are with the Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA 02446, {shiran27,cgc}@bu.edu.

Manuscript submitted October 17 2019

[25], [26]) and subsequently to explore the search space systematically for a better local optimum. However, none of the latter methods have been designed to address *distributed* cooperative multi-agent optimization problems and no convergence guarantees have been provided.

Building upon the centralized boosting function approach introduced specifically for coverage control problems in [12], we propose a distributed approach to solve general non-convex optimization problems associated with cooperative multi-agent systems. The boosting function approach can be thought of as a process where the local agent objective functions are temporarily altered by defining a set of auxiliary local objective functions whenever an equilibrium (i.e., a local optimum) is reached. Rather than switching the local objective functions, this process is carried out indirectly by systematically transforming the local gradient into a new *boosted gradient*. Hence, a *boosting function* is formally a transformation of local gradients into appropriate boosted gradients. Clearly, such transformations should always result in non-zero boosted gradients whenever the local gradients are zero so as to enable escaping the local optima. After the next equilibrium is reached following the boosted gradients, we switch back to the original local gradients (also called *normal gradients*). Subsequently, the gradient-based algorithm will converge to a new (potentially better) equilibrium point. Compared to methods where gradient components are randomly perturbed to escape local optima [13], the boosting function approach provides explicitly computed boosted gradients which ensure both escaping from the local optima and subsequent systematic exploration of the search space. As will be shown, such desirable qualities can be achieved by designing boosted gradients taking into account structural properties of the objective function as well as information such as the nature of the feasible space where agents can operate.

The contributions of this paper start with a proposed general-purpose Distributed Boosting Scheme (DBS) where each agent is allowed to asynchronously switch between a “boosting” and a “normal” mode independent of other agents and also without any global communication. Next, we propose a novel (generally applicable) *optimal variable step size* selection technique which ensures that the DBS converges. Although our motivation for both contributions mentioned above comes from the coverage control problem setting, they apply to a broad class of multi-agent systems, beyond coverage or consensus-like problems. As a final contribution, to illustrate the effectiveness of the DBS we use a class of multi-agent coverage control problems to provide more intuition by developing two new boosting function families not previously used in [12].

The paper is organized as follows. Section II introduces the general cooperative multi-agent optimization problem and the key concepts of boosted gradients and boosting schemes. Then, an optimal variable step size selection mechanism for gradient-based algorithms is presented along with related convergence proofs in Section III. In Section IV, we present an application of the boosting concepts to the class of multi-agent coverage control problems. Section IV-D presents simulation results illustrating the effectiveness of the distributed boosting framework and Section V concludes the paper including some future research directions.

II. Problem Formulation

We consider cooperative multi-agent optimization problems of the general form (see also [27]),

$$\mathbf{s}^* = \arg\max_{\mathbf{s}} \; H(\mathbf{s}), \tag{1}$$

where, $H : \mathbb{R}^{mN} \to \mathbb{R}$ is the *global objective* function and $\mathbf{s} = [s_1, s_2, \ldots, s_N] \in \mathbb{R}^{mN}$ is the *global state*. Here, $s_i \in \mathbb{R}^m$ represents the controllable *local state* of an agent $i \in \{1, 2, \ldots, N\}$. In this work, the global objective function $H(\mathbf{s})$ is not required to satisfy any linearity or convexity-related conditions.

To model inter-agent interactions, an undirected graph denoted by $\mathscr{G} = (\mathscr{V}, \mathscr{A})$ is used where $\mathscr{V} = \{1, 2, \ldots, N\}$ represent the set of N agents, and, $\mathscr{A}$ is the set of available communication links between those agents. The set of *neighbors* of an agent $i \in \mathscr{V}$ is denoted by $B_i = \{j : j \in \mathscr{V},\ (i, j) \in \mathscr{A}\}$ and the *closed neighborhood* of an agent i is defined as $\bar{B}_i = B_i \cup \{i\}$. It is assumed that each agent i shares its local state information s_i with its neighbors in B_i. As a result, agent i has knowledge of its *neighborhood state* $\bar{s}_i = \{s_j : j \in \bar{B}_i\}$.

In this problem setting, each agent (say i) is assumed to have a *local objective* function $H_i(\bar{s}_i)$ where $H_i : \mathbb{R}^{m|\bar{B}_i|} \to \mathbb{R}$ ($|\cdot|$ is the cardinality operator). Note that $H_i(\bar{s}_i)$ only depends on agent i's neighborhood state $\bar{s}_i$. The relationship between local and global objective functions is not restricted to any specific form except for the condition:

$$\frac{\partial H_i(\bar{s}_i)}{\partial s_i} = 0,\ \forall i \in \mathscr{V} \implies \nabla H(\mathbf{s}) = 0. \tag{2}$$

This condition clearly holds for any problem with a *separable* form [12] $H(\mathbf{s}) = H_i(\bar{s}_i) + H_i^c(s_i^c)$ where $H_i^c : \mathbb{R}^{m(N-1)} \to \mathbb{R}$ and $s_i^c = [s_1, s_2, \ldots, s_{i-1}, s_{i+1}, \ldots, s_N]$. Note that cooperative multi-agent systems which are inherently distributed (e.g., [1]) naturally have separable objective functions. Moreover, many problems of interest with an *additive* form [11] $H(\mathbf{s}) = \sum_{i=1}^{N} H_i(\bar{s}_i)$ also satisfy this condition (see also [28]).

Since H is not constrained to have any specific properties, complete solution techniques for (1) are limited to global optimization methods [29], [30]. However, the focus of this paper is on using a conventional gradient ascent approach so as to take advantage of its simplicity (in terms of analysis, computation, and on-line implementations) despite the obvious limitation of attaining only a local optimum. Moreover, another focus of this work is to develop a distributed optimization approach. Therefore, to solve (1), we consider the *distributed* scheme where each agent i updates its local state s_i according to

$$s_{i,k+1} = s_{i,k} + \beta_{i,k} d_{i,k}. \tag{3}$$

Here, $\beta_{i,k} \in \mathbb{R}$ is a step size and $d_{i,k} \triangleq \frac{\partial H_i(\bar{s}_{i,k})}{\partial s_i} \in \mathbb{R}^m$ denotes the locally available gradient of agent i.

We discuss the constrained case of (1) separately in Section III-C2 where $\mathbf{s}$ is constrained to a feasible space $\mathbf{F} \subseteq \mathbb{R}^{mN}$.

A. Escaping local optima using boosting functions

To overcome the problem of (3) converging to a local optimum when H is non-convex, the *boosting function* approach first proposed in [12] is adopted here. The attractiveness of this approach comes from the fact that it enables agents to systematically escape local optima without compromising the simplicity of the proposed solution technique. The main idea of this approach is to temporarily replace the local objective function $H_i(\bar{s}_i)$ whenever an equilibrium is reached with an auxiliary objective function $\hat{H}_i(\bar{s}_i)$. Since this is equivalent to replacing the *normal gradient* d_i by a *boosted gradient* $\hat{d}_i = \frac{\partial \hat{H}_i(\bar{s}_i)}{\partial s_i}$ in (3), we focus instead on constructing $\hat{d}_i$ directly.

A *boosting function* f_i can be thought of as a transformation of an associated normal gradient d_i which results in a boosted gradient $\hat{d}_i = f_i(d_i)$. In particular, this transformation takes place at an equilibrium point (where $d_i = 0$) and should result in a non-zero boosted gradient $\hat{d}_i = f_i(0) \neq 0$ which, therefore, forces agent i to move in a direction determined by the boosting function and to explore the feasible space further. Subsequently, when a new equilibrium point is reached (i.e., when $\hat{d}_i = 0$), the agent reverts to the normal gradient d_i and the gradient-based algorithm converges to a new (potentially better and never worse) equilibrium point.

The key to boosting functions is that they are selected to exploit the structure of: (i) the objective functions $H(\mathbf{s})$ and $H_i(\bar{s}_i)$, (ii) the gradient expression d_i, (iii) the feasible space $\mathbf{F}$, and, (iv) the agent state trajectory history. Unlike various forms of randomized state perturbations away from their current equilibrium [13], [14], boosting functions provide a formal rational systematic transformation process where the boosting function f_i depends on the specific problem type. Details on boosting functions and their design process are discussed in Section IV. In what follows, we present a general-purpose boosting function choice to provide insight into boosting functions in a generic setting.

In many multi-agent optimization problems, local optima arise when a cluster of agents provides a reasonably high performance by maintaining their local states in close proximity while completely ignoring globally dispersed state configurations. In such a case, a boosting function that enhances a separation among local states is a natural choice, especially suited for applications like coverage control, formation control, monitoring, consensus and transportation. In fact, for coverage control problems, such a boosting function has already been proven to be effective in [12]. Therefore, in a generic setting, a candidate boosted gradient $\hat{d}_i = f_i(d_i)$ for agent i can be obtained by letting $\psi_{ij} = (s_i - s_j)$ and defining $\hat{d}_i = \nabla_{\psi_{ij}} H_i(\bar{s}_i)$ where its l^{th} component is

$$\hat{d}_i^l = \frac{\partial H_i(\bar{s}_i)}{\partial \psi_{ij}^l} = \underbrace{\frac{\partial H_i(\bar{s}_i)}{\partial s_i^l}}_{= d_i^l} \frac{\partial s_i^l}{\partial \psi_{ij}^l} + \underbrace{\frac{\partial H_i(\bar{s}_i)}{\partial s_j^l}}_{\triangleq d_{ji}^l} \frac{\partial s_j^l}{\partial \psi_{ij}^l}. \tag{4}$$

Now, by replacing $\frac{\partial s_i^l}{\partial \psi_{ij}^l}$ and $\frac{\partial s_j^l}{\partial \psi_{ij}^l}$ with scalar parameters α_{ij} and η_{ij}, an entire *family of boosting functions* can be obtained as $\hat{d}_i = f_i(d_i) = \alpha_{ij} d_i + \eta_{ij} d_{ji}$ where $d_{ji} = \frac{\partial H_i(\bar{s}_i)}{\partial s_j}$ (see also (52) and (53)). Note that setting $\alpha_{ij} = 1$ and $\eta_{ij} = -1$ gives an interesting boosting function choice of the form $\hat{d}_i = f_i(d_i) = d_i - d_{ji}$. Since d_{ji} represents the direction towards which agent j should move to increase H_i, this is clearly an intuitive general choice for a boosting function at i. Details on selecting boosting functions along with some guidelines can be found in [28].

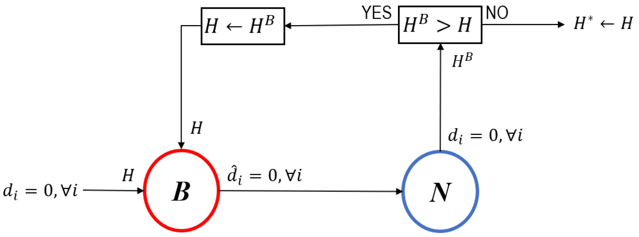


Fig. 1: A centralized boosting scheme (CBS).

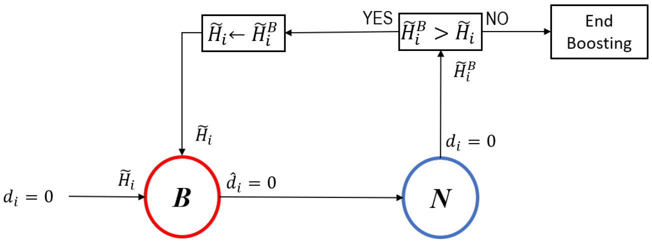


Fig. 2: A distributed boosting scheme (DBS) asynchronously applied by each agent $i = 1, \ldots, N$.

1) Boosting schemes: An agent i is said to be in the *Boosting Mode* when it is following the boosted gradient direction $\hat{d}_i$ where its state updates take the form

$$s_{i,k+1} = s_{i,k} + \beta_{i,k} \hat{d}_{i,k}. \tag{5}$$

Similarly, when an agent i is following the "normal" gradient direction $d_{i,k}$ as in (3), it is said to be in the *Normal Mode*. When developing an optimization scheme to solve (1), we need a proper mechanism, referred to as a *Boosting Scheme,* to switch the agents between normal and boosting modes.

A centralized boosting scheme (CBS) is outlined in Fig. 1, where the boosting mode is denoted by $\boldsymbol{B}$ and the normal mode is denoted by $\boldsymbol{N}$. In a CBS, all agents are synchronized to operate in the same mode. In Fig. 1, H denotes the global objective function value which is initially stored by all agents the first time mode $\boldsymbol{B}$ is entered when $d_i = 0$ for all $i \in \mathscr{V}$. After $\hat{d}_i = 0$ for all $i \in \mathscr{V}$, the agents re-enter mode $\boldsymbol{N}$ and, when a new equilibrium is reached, the new post-boosting value of the global objective function $H(\mathbf{s})$ is denoted by H^B. If $H^B > H$, an improved equilibrium point is attained and the process repeats by re-entering mode $\boldsymbol{B}$ with the new value H^B. The process is complete when this centralized controller fails to improve $H(\mathbf{s})$, i.e., when $H^B \leq H$.

This CBS was used in [12] with appropriately defined boosting functions in mode $\boldsymbol{B}$ to obtain improved performance for a variety of multi-agent coverage control problems. However, there has been no formal proof to date that this process converges. Moreover, our goal is to develop a *Distributed Boosting Scheme* (DBS) where each agent can independently switch between modes $\boldsymbol{B}$ and $\boldsymbol{N}$ at any time. Such a scheme

(i) improves the scalability of the system, (ii) eliminates the requirement of a centralized controller, (iii) reduces computational and communication costs, and, (iv) can potentially improve convergence times. Furthermore, this is a natural approach in problems such as coverage control [1], where the original problem is inherently distributed.

A simple DBS version of Fig. 1 is shown in Fig. 2 where local use of the global objective H is now replaced by a local estimate of H, denoted by $\bar{H}_i$, which will be formally introduced later. One can see that convergence of the DBS is far from obvious since agents may be at different modes at any time instant and, as their states change, the interaction among agents could lead to oscillatory behavior. Note that the notion of convergence involves not only the existence of equilibria such that $d_i = 0$ or $\hat{d}_i = 0$, but also a guarantee that the condition $H^B \leq H$ is eventually satisfied. We will show that a key to guaranteeing convergence is a process for *optimally selecting a variable step size* $\beta_{i,k}$ in (3) and (5).

2) Convergence criteria: When a DBS is considered, the decentralized nature of agent behavior causes agents to switch between normal and boosting modes independently and asynchronously from each other (unlike a CBS). As a result, at a given time instant, a subset of the agents will be in normal mode (following (3)) while others are in boosting mode (following (5)). This creates a partition of the complete agent set $\mathscr{V}$ into two agent sets henceforth denoted by $\mathscr{N}$ and $\mathscr{B}$ respectively.

Let us define the *extended neighborhood* of an agent i as $\tilde{B}_i \triangleq \cup_{j \in \bar{B}_i} \bar{B}_j$. For any agent $i \in \mathscr{V}$, the following conditions are defined as the *convergence criteria*:

$$\lim_{k\to\infty} d_{i,k} = 0 \text{ when } \tilde{B}_i \subseteq \mathscr{N}, \tag{6}$$

$$\lim_{k\to\infty} d_{i,k} = 0 \text{ when } i \in \mathscr{N}, \tilde{B}_i \cap \mathscr{B} \neq \emptyset, \tag{7}$$

$$\lim_{k\to\infty} \hat{d}_{i,k} = 0 \text{ when } i \in \mathscr{B}, \tilde{B}_i \cap \mathscr{B} \neq \emptyset. \tag{8}$$

These convergence criteria enforce the capability of an agent i to escape its current mode (normal or boosting) irrespective of the surrounding neighbor mode partitions $\tilde{B}_i \cap \mathscr{N}$ and $\tilde{B}_i \cap \mathscr{B}$. Since boosting will only continue as long as there is a gain from the boosting stages (i.e., "$\bar{H}_i^B > \bar{H}_i$" in Fig. 2), it is clear how these criteria can lead all agents to terminate their boosting stages (i.e., to reach the "End Boosting" state).

Upon this termination, criterion (6) reapplies and guarantees achieving $d_{i,k} \to 0$ for all $i \in \mathscr{V}$, which will directly imply $\nabla H(\mathbf{s}_k) \to 0$ (from (2)). Therefore, convergence to a stationary solution of (1) is achieved (again, not necessarily a global optimum). Finally, note that the criterion (6) applies to the convergence of any gradient-based method where boosting is not used.

III. Convergence Analysis Through Optimal Variable Step Sizes

This section proposes a variable step size scheme which guarantees the convergence criteria (6)-(8) required when a general problem of the form (1) is solved using (3) and (5). Our main results are Theorem 1 (which guarantees (6)) and Theorem 2 (which guarantees (7) and (8)). These depend on some assumptions which are presented first, starting with the nature of the local objective functions.

Assumption 1: Any local objective function $H_i(\bar{s}_i)$, $i \in \mathscr{V}$, satisfies the following conditions:

1) $H_i(\cdot)$ is continuously differentiable and its gradient $\nabla H_i(\cdot)$ is Lipschitz continuous (i.e., $\exists K_{1i}$ such that $\forall x, y \in \mathbb{R}^{m|\bar{B}_i|}$, $\|\nabla H_i(x) - \nabla H_i(y)\| \leq K_{1i}\|x-y\|$).
2) $H_i(\cdot)$ is a non-negative function with a finite upper bound H_{UB}, i.e., $H_i(x) < H_{UB} < \infty$, $x \in \mathbb{R}^{m|\bar{B}_i|}$.

Through the relationship between local and global objective functions, this assumption will propagate to the global objective function. However, for this work, Assumption 1 is sufficient.

A. Convergence for agents $i \in \mathscr{V}$ such that $\tilde{B}_i \subseteq \mathscr{N}$

We begin by developing an optimal variable step size scheme for agents $i \in \mathscr{V}$ such that $\tilde{B}_i \subseteq \mathscr{N}$, i.e., all agents in the extended neighborhood are in normal mode - following (3). The respective convergence criterion for this case is (6). For notational convenience, let $\mathbf{q}_i = \{1, 2, \dots, q_i\}$ with $q_i = |\bar{B}_i|$ representing an ordered (re-indexed) version of the closed neighborhood set $\bar{B}_i$. For this situation, agent i's neighborhood state update equation can be expressed as $\bar{s}_{i,k+1} = \bar{s}_{i,k} + \bar{\beta}_{i,k}\bar{\mathbf{d}}_{i,k}$ by combining (3) for all $j \in \bar{B}_i$. Here, $\bar{s}_{i,k+1}$, $\bar{s}_{i,k}$ and $\bar{\mathbf{d}}_{i,k}$ are mq_i-dimensional column vectors; equivalently, they may be thought of as $q_i \times 1$ block-column matrices with their j^{th} block (of size $\mathbb{R}^{m\times 1}$, and $j \in \mathbf{q}_i$) being, $s_{j,k+1}$, $s_{j,k}$ and $d_{j,k}$ respectively. Accordingly, $\bar{\beta}_{i,k}$ is a $q_i \times q_i$ block-diagonal matrix, where its j^{th} block on the diagonal (of size $m \times m$ and $j \in \mathbf{q}_i$) is $\beta_{j,k}I_m$; I_m is the $m \times m$ identity matrix and $\beta_{j,k} \in \mathbb{R}$ is the (scalar) step size of agent j.

The following lemma provides a modified version of the widely used descent lemma [31] so that it can be applied to analyze maximization problems such as (1).

Lemma 1: (Ascent lemma) For a function $f : \mathbb{R}^n \to \mathbb{R}$, if the Lipschitz continuity constant of ∇f is L, then, $\forall x, y \in \mathbb{R}^n$,

$$f(x+y) \geq f(x) + y^T\nabla f(x) - \frac{L}{2}\|y\|^2. \tag{9}$$

Proof: Consider a function $g = -f : \mathbb{R}^n \to \mathbb{R}$. Then, the Lipschitz continuity constant of ∇g will also be L. We can now apply the usual descent lemma [31] to the function g (to compare $g(x+y)$ and $g(x)$). Then, using $g = -f$, $\forall x, y \in \mathbb{R}^n$,

$$-g(x+y) \geq -g(x) - y^T\nabla g(x) - \frac{L}{2}\|y\|^2,$$

and the result follows. ■

Now, under Assumption 1, Lemma 1 can be applied to any local objective function $H_i(\bar{s}_{i,k})$ for the aforementioned neighborhood state update $\bar{s}_{i,k+1} = \bar{s}_{i,k} + \bar{\beta}_{i,k}\bar{\mathbf{d}}_{i,k}$ as follows:

$$\begin{aligned} H_i(\bar{s}_{i,k+1}) &\geq H_i(\bar{s}_{i,k}) + (\bar{\beta}_{i,k}\bar{\mathbf{d}}_{i,k})^T\nabla H_i(\bar{s}_{i,k}) - \frac{K_{1i}}{2}\|\bar{\beta}_{i,k}\bar{\mathbf{d}}_{i,k}\|^2 \\ &= H_i(\bar{s}_{i,k}) + \sum_{j\in\bar{B}_i}\left[\beta_{j,k}d_{j,k}^T d_{ji,k} - \frac{K_{1i}}{2}\beta_{j,k}^2\|d_{j,k}\|^2\right] \\ &= H_i(\bar{s}_{i,k}) + \sum_{j\in\bar{B}_i}\Delta_{ji,k}, \end{aligned} \tag{10}$$

with

$$\Delta_{ji,k} \triangleq \beta_{j,k} d_{j,k}^T d_{ji,k} - \frac{K_{1i}}{2}\beta_{j,k}^2 \|d_{j,k}\|^2 \in \mathbb{R}, \tag{11}$$

$$d_{ji,k} \triangleq \nabla_j H_i(\bar{s}_{i,k}) = \frac{\partial H_i(\bar{s}_{i,k})}{\partial s_j} \in \mathbb{R}^m. \tag{12}$$

The term $d_{ji,k}$ in (12) gives the sensitivity of agent i's local objective H_i to the local state s_j of agent $j \in \bar{B}_i$. Also, K_{1i} is the Lipschitz constant corresponding to ∇H_i. Note that the term $\Delta_{ji,k}$ in (11) depends on the step size $\beta_{j,k}$ which is selected by agent $j \in \bar{B}_i$.

In (10), each $\Delta_{ji,k}$ term can be thought of as a contribution from a neighboring agent j to agent i, so as to improve (increase) H_i. However, in order for any agent i to know its contribution to agent $j \in \bar{B}_i$ (i.e., $\Delta_{ij,k}$), the following assumption is required.

Assumption 2: Any agent $i \in \mathscr{V}$ has knowledge of the cross-gradient terms $d_{ij,k}$, the local Lipschitz constants K_{1j}, and the objective function values $H_j(\bar{s}_{j,k})$ at the kth update instant.

This assumption is consistent with our concept of neighborhood, where neighbors share information through communication links. Note that when the form of the local objective functions H_i is identical and all pairs (H_i, H_j), $j \in B_i$, have a symmetric structure, Assumption 2 holds without any need for additional communication sessions. In fact, many cooperative multi-agent optimization problems inherently have this property of symmetry including the class of multi-agent coverage control problems which will be discussed in Section IV (see Lemma 7).

We now define a *neighborhood objective* function $\tilde{H}_i(\tilde{s}_{i,k})$ for any $i \in \mathscr{V}$, where $\tilde{H}_i : \mathbb{R}^{m|\tilde{B}_i|} \rightarrow \mathbb{R}$ and $\tilde{s}_{i,k} = \{s_j : j \in \tilde{B}_i\}$, as follows:

$$\tilde{H}_i(\tilde{s}_{i,k}) = \sum_{j\in\bar{B}_i} H_j(\bar{s}_{j,k}). \tag{13}$$

This neighborhood objective function can be viewed as agent i's estimate of the total contribution of agents in $\bar{B}_i$ towards the global objective function. These functions play an important role in the DBS because a distributed scheme comes at the cost of each agent losing the global information $H(\mathbf{s})$. Recall that in the CBS of Fig. 1, $H(\mathbf{s})$ plays a crucial role in the "$H^B > H$" block. In contrast, in a DBS, each agent i uses a neighborhood objective function $\tilde{H}_i$ as a means of locally estimating the global objective function value (see "$\tilde{H}_i^B > \tilde{H}_i$" block in Fig. 2). However, as seen in the ensuing analysis, the form of $\tilde{H}_i$ is not limited to (13) - it can take any appropriate form.

Remark 1: In some problems, if the global and local objective functions are not directly related in an additive manner, then $\tilde{H}_i(\tilde{s}_{i,k}) = \sum_{j\in\bar{B}_i} w_{ij} H_j(\bar{s}_{j,k})$ can be used as a candidate for the neighborhood objective function. Here, $\{w_{ij} \in \mathbb{R}_{\geq 0} : j \in \bar{B}_i\}$ represents a set of weights (scaling factors). All results presented in this section can be generalized to such neighborhood objective functions as well.

Enabled by the fact that $\tilde{B}_i \subseteq \mathscr{N} = \emptyset$, applying (10) to any agent $j \in \bar{B}_i$ gives $H_j(\bar{s}_{j,k+1}) \geq H_j(\bar{s}_{j,k}) + \sum_{l\in\bar{B}_j} \Delta_{lj,k}$. Summing both sides of this relationship over all $j \in \bar{B}_i$ and using the definition in (13) yields

$$\tilde{H}_i(\tilde{s}_{i,k+1}) \geq \tilde{H}_i(\tilde{s}_{i,k}) + (\tilde{\Delta}_{i,k} + Q_{i,k}), \tag{14}$$

where we define

$$\tilde{\Delta}_{i,k} \triangleq \sum_{j\in\bar{B}_i} \Delta_{ij,k}, \text{ and,} \tag{15}$$

$$Q_{i,k} \triangleq \sum_{j\in B_i} (\Delta_{jj,k} + \Delta_{ji,k} + \sum_{l\in B_j-\{i\}} \Delta_{lj,k}). \tag{16}$$

Note that $\tilde{\Delta}_{i,k}$ in (15) is a function of terms $\Delta_{ij,k}$ (and not $\Delta_{ji,k}$) which are locally available and controlled by agent i, i.e., via terms $\beta_{i,k}, d_{i,k}$ and $d_{ij,k}, \forall j \in \bar{B}_i$. However, agent i does not have any control over $Q_{i,k}$ in (16), as this strictly depends (through (11)) on the step sizes of agent i's neighbors in its *extended neighborhood* $\tilde{B}_i$ (i.e., $\beta_{j,k}, \forall j \in \tilde{B}_i - \{i\}$).

Nonetheless, (14) implies that the neighborhood objective function $\tilde{H}_i(\tilde{s}_{i,k})$ can be increased by at least $(\tilde{\Delta}_{i,k} + Q_{i,k})$ at any update instant k. Thus, to maximize the gain in $\tilde{H}_i(\tilde{s}_{i,k})$, agent i's step size $\beta_{i,k}$ is selected according to the following *auxiliary problem*:

$$\begin{aligned} \beta_{i,k}^* = \arg\max_{\beta_{i,k}} \quad & \tilde{\Delta}_{i,k} \\ \text{subject to} \quad & \tilde{\Delta}_{i,k} > 0. \end{aligned} \tag{17}$$

Lemma 2: The solution to the auxiliary problem (17) is

$$\beta_{i,k}^* = \frac{1}{\sum_{j\in\bar{B}_i} K_{1j}} \frac{d_{i,k}^T \sum_{j\in\bar{B}_i} d_{ij,k}}{\|d_{i,k}\|^2}. \tag{18}$$

Proof: Using (11) and (15), $\tilde{\Delta}_{i,k}$ can be written as

$$\tilde{\Delta}_{i,k} = \beta_{i,k} d_{i,k}^T \sum_{j\in\bar{B}_i} d_{ij,k} - \beta_{i,k}^2 \|d_{i,k}\|^2 \frac{\sum_{j\in\bar{B}_i} K_{1j}}{2}. \tag{19}$$

Note the quadratic and concave nature of $\tilde{\Delta}_{i,k}$ with respect to agent i's step size $\beta_{i,k}$. Therefore, using the KKT conditions [31], the optimal $\beta_{i,k}$ can be directly obtained as (18). Let us denote the optimal objective function value as $\tilde{\Delta}_{i,k}^*$. It is easy to show that $\beta_{i,k}^*$ in (18) is feasible (i.e., $\tilde{\Delta}_{i,k}^* > 0$) as long as $\beta_{i,k}^* \neq 0$. ■

Remark 2: The extreme situation where $\beta_{i,k}^* = 0$ occurs when $\sum_{j\in\bar{B}_i} d_{ij,k} = 0$. However, since this "pathological situation" can be detected by agent i, the agent can consider two options: (i) Use a reduced neighborhood $\bar{B}_i^1 \subset \bar{B}_i$ to calculate $\beta_{i,k}^*$ so that $\beta_{i,k}^* \neq 0$, hence $\tilde{\Delta}_{i,k}^* > 0$, or (ii) Use the weighted form of (13) (see Remark 1) and manipulate the weight factors $\{w_{ij} : j \in \bar{B}_i\}$ so as to get a step size $\beta_{i,k}^* \neq 0$ (e.g., enforcing $w_{ij} = 0, \forall j \ni d_{i,k}^T d_{ij} < 0$ will give $\beta_{i,k} > 0$, hence $\tilde{\Delta}_{i,k}^* > 0$). Therefore, in the following analysis, we omit this pathological situation by assuming $\sum_{j\in\bar{B}_i} d_{ij,k} \neq 0$ (which implies $\beta_{i,k}^* \neq 0$).

By substituting (18) in $\tilde{\Delta}_{i,k}$ given in (19), we can obtain an explicit expression for $\tilde{\Delta}_{i,k}^*$ as $\tilde{\Delta}_{i,k}^* = \frac{1}{2}\beta_{i,k}^* d_{i,k}^T \sum_{j\in\bar{B}_i} d_{ij,k}$. From this result and in view of Remark 2, it is clear that $\tilde{\Delta}_{i,k}^* \rightarrow 0$ if an only if $d_{i,k} \rightarrow 0$.

Next, regarding the term $Q_{i,k}$ in (16) over which agent i does not have any control, the following lemma can be established.

Lemma 3: The term $Q_{i,k}$ can be expressed as

$$Q_{i,k} = \sum_{j\in B_i} (\tilde{\Delta}_{j,k} + \sum_{l\in B_j-\{i\}} \left[\Delta_{lj,k} - \Delta_{jl,k}\right]). \tag{20}$$

Further, if $B_i = \bar{B}_j - \{i\}$, then under (18), $Q_{i,k} > 0$.

Proof: In (16), let us add and subtract $\sum_{l \in B_j - \{i\}} \Delta_{jl,k}$ to the inner terms of the main summation. Then, using the definition (15), the expression in (20) is obtained. To prove the second part, note that the first inner term of the main summation of (20) (i.e., $\tilde{\Delta}_{j,k}$) is always positive under the optimal step size given in (18). Let us then consider the net effect of the second inner term of $Q_{i,k}$, denoted by $Q'_{i,k}$, where we have

$$Q'_{i,k} = Q_{i,k} - \sum_{j \in B_i} \tilde{\Delta}_{j,k} = \sum_{j \in B_i} \sum_{l \in B_j - \{i\}} \left[\Delta_{lj,k} - \Delta_{jl,k} \right].$$

Using the fact that $\Delta_{lj,k} - \Delta_{jl,k} = 0$ when $l = j$, we can add a dummy term into the inner summation to get

$$Q'_{i,k} = \sum_{j \in B_i} \sum_{l \in \bar{B}_j - \{i\}} \left[\Delta_{lj,k} - \Delta_{jl,k} \right] = \sum_{j \in B_i} \sum_{l \in B_i} \left[\Delta_{lj,k} - \Delta_{jl,k} \right],$$

where the last step follows from the assumption $B_i = \bar{B}_j - \{i\}$. Observing that the two running variables l, j in the summations above are interchangeable, we get $Q'_{i,k} = 0$. This implies that under (18), $Q_{i,k} = \sum_{j \in B_i} \tilde{\Delta}_{j,k} > 0$. ■

We now make the following assumption regarding $Q_{i,k}$.

Assumption 3: Consider the sum,

$$\tilde{Q}_{i,k} = \sum_{l=k-T_i}^{k} Q_{i,l}, \tag{21}$$

such that $0 \leq T_i \leq k$. Then, $\exists T_i < \infty$ such that $\tilde{Q}_{i,k} \geq 0$.

When the graph $\mathscr{G}(\mathscr{V}, \mathscr{A})$ is complete, the condition $B_i = \bar{B}_j - \{i\}$ in Lemma 3 is true for all $i \in \mathscr{V}$. In such cases, Assumption 3 is immediately satisfied with $T_i = 1, \forall i \in \mathscr{V}$. On the other hand, when the graph $\mathscr{G}(\mathscr{V}, \mathscr{A})$ is sparse enough, it can be considered as a collection of fully connected sub-graphs (exploiting the partitioned nature of local objective functions $H_i(\bar{s}_i)$). Then, Assumption 3 also holds with $T_i = 1, \forall i \in \mathscr{V}$. More generally, when each agent selects its step size according to (18), it ensures that $\tilde{\Delta}^*_{i,k} > 0$. In addition, $\Delta_{ii,k} > 0$ whenever the step size $\beta_{i,k}$ is positive. The assumption is further supported by the fact that each $Q_{i,k}$ in $\tilde{Q}_{i,k}$ is also a summation of $\Delta_{jj,k}$, $\Delta_{ji,k}$ and $\Delta_{lj,k}$ terms (noting in particular the positive first terms in (16) as well as in (20)). Moreover, it is locally verifiable if the agent communicates with its neighbors. In practice, we have never seen this assumption violated over extensive simulation examples (see Fig. 5 in Section IV and accompanying discussion).

To establish the convergence proof in Theorem 1, we need to make one final assumption.

Assumption 4: For all $i \in \mathscr{V}$, there exists a function $\Psi_{i,k}$ and a finite positive number ε such that $\Psi_{i,k} \geq \varepsilon > 0$ and

$$\begin{cases} 0 \leq \Psi_{i,k} \|d_{i,k}\|^2 \leq \tilde{\Delta}^*_{i,k} + \tilde{Q}_{i,k}, \text{ when } \tilde{Q}_{i,k} > 0,\ \tilde{\Delta}^*_{i,k} > 0, & (22) \\ 0 \leq \Psi_{i,k} \|d_{i,k}\|^2 \leq \tilde{\Delta}^*_{i,k}, \text{ when } \tilde{\Delta}^*_{i,k} > 0. & (23) \end{cases}$$

This assumption is trivial because whenever the optimal step size in (18) is used, $0 < \tilde{\Delta}^*_{i,k}$, hence, for some $1 < K_2$, $\Psi_{i,k} = \tilde{\Delta}^*_{i,k} / (K_2 \|d_{i,k}\|^2)$ is a candidate function for both cases (22) and (23) that satisfies the requirement $\Psi_{i,k} \geq \varepsilon > 0$ for all k.

Now, the main convergence theorem can be established.

Theorem 1: For all $i \in \mathscr{V}$ such that $\bar{B}_i \subseteq \mathscr{N}$, under Assumptions 1-4, the step size selection in (18) guarantees the convergence criterion (6), i.e., $\lim_{k \to \infty} d_{i,k} = 0$.

Proof: By Assumption 3, a T_i value can be defined for $\tilde{Q}_{i,k}$ at each k. Consider a sequence of consecutive discrete update instants $\{k_1 + 1, \ldots, k'_1\}$ (for short, we use the notation $(k_1, k'_1]$), where, $T_i = k'_1 - k_1$ is associated with $\tilde{Q}_{i,k'_1}$ and $T_i > k - k_1$ applies to all $\tilde{Q}_{i,k}$, $k \in (k_1, k'_1 - 1]$. This means $0 < \sum_{k=k_1+1}^{k'_1} Q_{i,k}$ and $0 \geq \sum_{k=k_1+1}^{l} Q_{i,k}$, $\forall l \in (k_1, k'_1 - 1]$. In addition, by Lemma 2, $0 < \tilde{\Delta}^*_{i,k}$ $\forall k \in (k_1, k'_1]$. Thus, $0 < \sum_{k=k_1+1}^{k'_1} (\tilde{\Delta}^*_{i,k} + Q_{i,k})$. By summing up both sides of (14) over all update steps $k \in (k_1, k'_1]$ yields

$$\tilde{H}_i(\tilde{s}_{i,k'_1+1}) \geq \tilde{H}_i(\tilde{s}_{i,k_1+1}) + \sum_{k=k_1+1}^{k'_1} (\tilde{\Delta}^*_{i,k} + Q_{i,k}). \tag{24}$$

Similarly, using Assumption 4 and summing both sides of (23) over all $k \in (k_1, k'_1 - 1]$ and using (22) for $k = k'_1$ yields

$$0 \leq \sum_{k=k_1+1}^{k'_1} \Psi_{i,k} \|d_{i,k}\|^2 \leq \sum_{k=k_1+1}^{k'_1} (\tilde{\Delta}^*_{i,k} + Q_{i,k}). \tag{25}$$

By Assumption 3, the length of the chosen interval $(k_1, k'_1]$ is always finite. Therefore, any $\{1, \ldots, k_2\}$ with $k_2 < \infty$ can be decomposed into a sequence of similar sub-intervals: $\{(k_{11}, k'_{11}], (k_{12}, k'_{12}], \ldots, (k_{1L}, k'_{1L}]\}$ where $k_{11} = 0$, $k'_{1i} = k_{1(i+1)}$ $\forall i \in (0, L]$. If k_2 is such that $k'_{1L} < k_2$ (which happens if $0 > \sum_{k=k'_{1L}+1}^{k_2} Q_{i,k}$), Assumption 3 implies that there exists some k'_2 such that $k_2 < k'_2 < \infty$ which satisfies $0 < \sum_{k=k'_{1L}+1}^{k'_2} Q_{i,k}$ (i.e., $(k'_{1L}, k'_2]$ is the new last sub-interval of $(0, k'_2]$). Then, by writing the respective expressions in (24) and (25) for each such sub-interval of the complete interval $(0, k'_2]$ and summing both sides over all k yields

$$\tilde{H}_i(\tilde{s}_{i,k'_2+1}) \geq \tilde{H}_i(\tilde{s}_{i,1}) + \sum_{k=1}^{k'_2} (\tilde{\Delta}^*_{i,k} + Q_{i,k}), \tag{26}$$

$$0 \leq \sum_{k=1}^{k'_2} \Psi_{i,k} \|d_{i,k}\|^2 \leq \sum_{k=1}^{k'_2} (\tilde{\Delta}^*_{i,k} + Q_{i,k}), \tag{27}$$

respectively. Using Assumption 1 in (26) gives $|\bar{B}_i| H_{UB} \geq \tilde{H}_i(\tilde{s}_{i,k'_2+1}) - \tilde{H}_i(\tilde{s}_{i,1}) \geq \sum_{k=1}^{k'_2} (\tilde{\Delta}^*_{i,k} + Q_{i,k})$. Combining this with (27) yields

$$\sum_{k=1}^{k'_2} \Psi_{i,k} \|d_{i,k}\|^2 \leq |\bar{B}_i| H_{UB}. \tag{28}$$

By Assumption 1, the term $|\bar{B}_i| H_{UB}$ in (28) is a finite positive number. Also, by Assumption 4, $\Psi_{i,k} \geq \varepsilon > 0$, $\forall k$. Therefore, taking limits of the above expression as $k'_2 \to \infty$ implies the convergence criterion in (6) as long as the optimal step sizes given by (18) are used. ■

B. Convergence for agents i such that $\bar{B}_i \cap \mathscr{B} \neq \emptyset$

In this case, at least some of the agents in $\bar{B}_i$ are in boosting mode, following (5). Using the same approach as in Section III-A, we seek an optimal variable step size selection scheme similar to (18) so as to ensure the convergence criteria in (7) and (8).

Compared to (10), now the ascent lemma relationship for $H_i(\bar{s}_{i,k})$ takes the form:

$$H_i(\bar{s}_{i,k+1}) \geq H_i(\bar{s}_{i,k}) + \sum_{j \in \bar{B}_i \cap \mathscr{N}} \Delta_{ji,k} + \sum_{j \in \bar{B}_i \cap \mathscr{B}} \hat{\Delta}_{ji,k}, \tag{29}$$

where $\Delta_{ji,k}$ for $j \in \mathscr{N}$ is the same as (11) and we set

$$\hat{\Delta}_{ji,k} = \beta_{j,k} \hat{d}_{j,k}^T d_{ji,k} - \frac{K_{1i}}{2} \beta_{j,k}^2 \|\hat{d}_{j,k}\|^2 \in \mathbb{R}. \tag{30}$$

Then, the ascent lemma for neighborhood objective function $\tilde{H}_i(\tilde{s}_{i,k})$ can be expressed as

$$\tilde{H}_i(\tilde{s}_{i,k+1}) \geq \tilde{H}_i(\tilde{s}_{i,k}) + (\tilde{\Delta}_{i,k} + Q_{i,k}), \tag{31}$$

with

$$\tilde{\Delta}_{i,k} \triangleq 1_{\{i \in \mathscr{N}\}} [\sum_{j \in \bar{B}_i} \Delta_{ij(k)}] + 1_{\{i \in \mathscr{B}\}} [\sum_{j \in \bar{B}_i} \hat{\Delta}_{ij(k)}], \tag{32}$$

$$\begin{aligned} Q_{i,k} \triangleq \sum_{j \in B_i} (1_{\{j \in \mathscr{N}\}} [\Delta_{jj,k} + \Delta_{ji,k}] + 1_{\{j \in \mathscr{B}\}} [\hat{\Delta}_{jj,k} + \hat{\Delta}_{ji,k}] \\ + \sum_{l \in \{B_j - \{i\}\}} [1_{\{l \in \mathscr{N}\}} \Delta_{lj,k} + 1_{\{l \in \mathscr{B}\}} \hat{\Delta}_{lj(k)}]), \end{aligned} \tag{33}$$

where $1_{\{\cdot\}}$ is the usual indicator function. Under this new $\tilde{\Delta}_{i,k}$ in (32), the same auxiliary problem as in (17) is used to determine the step size $\beta_{i,k}^*$ to optimally increase the neighborhood cost function $\tilde{H}_i(\tilde{s}_k)$. The corresponding solution (i.e., $\beta_{i,k}^*$) is given in the following lemma.

Lemma 4: The solution to the auxiliary problem (17) with $\tilde{\Delta}_{i,k}$ given in (32) is

$$\beta_{i,k}^* = \begin{cases} \frac{1}{\sum_{j \in \bar{B}_i} K_{1j}} \frac{d_{i,k}^T (\sum_{j \in \bar{B}_i} d_{ij,k})}{\|d_{i,k}\|^2} & \text{when } i \in \mathscr{N}, \\ \frac{1}{\sum_{j \in \bar{B}_i} K_{1j}} \frac{\hat{d}_{i,k}^T (\sum_{j \in \bar{B}_i} d_{ij,k})}{\|\hat{d}_{i,k}\|^2} & \text{when } i \in \mathscr{B}. \end{cases} \tag{34}$$

Proof: The proof follows the same steps as that of Lemma 2 and is, therefore, omitted. ∎

Note that the step size selection criterion given in (34) (for an agent i) does not depend on its neighbors' modes. Therefore, it offers a generalization of (18). However, $\beta_{i,k}^*$ now depends on agent i's own mode. This is due to the fact that the selection of $\beta_{i,k}^*$ allows agent i to maximize the increment in the neighborhood objective function $\tilde{H}_i(\tilde{s}_i)$ which is defined in (13) independently from the boosting process. Therefore, the use of $\beta_{i,k}^*$ provides a regulation mechanism for the state update steps (especially during the boosting mode).

To establish the convergence criteria (7) and (8), Assumptions 1, 2 and 3 are still required. Note that Assumption 3 should now be considered under the new expression for $Q_{i,k}$ in (33); its justification is similar as before. Moreover, a generalized version of Lemma 3 is given as follows.

Lemma 5: The term $Q_{i,k}$ in (33) can be expressed as,

$$\begin{aligned} Q_{i,k} = \sum_{j \in B_i} (\tilde{\Delta}_{j,k} + \sum_{l \in B_j - \{i\}} [1_{\{l \in \mathscr{N}\}} \Delta_{lj,k} - 1_{\{j \in \mathscr{N}\}} \Delta_{jl,k} \\ + 1_{\{l \in \mathscr{B}\}} \hat{\Delta}_{lj,k} - 1_{\{j \in \mathscr{B}\}} \hat{\Delta}_{jl,k}]). \end{aligned} \tag{35}$$

Further, if $B_i = \bar{B}_j - \{i\}$, then under (34), $Q_{i,k} > 0$.

Proof: The proof follows the same steps as that of Lemma 3 and is, therefore, omitted. ∎

Finally, before proceeding to Theorem 2, the previous Assumption 4 needs to be modified as follows so that it incorporates the possibility that $i \in \mathscr{B}$.

Assumption 5: For all $i \in \mathscr{V}$, there exists a function $\Psi_{i,k}$ and a finite positive number ε such that $\Psi_{i,k} \geq \varepsilon > 0$ and,
if $i \in \mathscr{N}$:

$$\begin{cases} 0 \leq \Psi_{i,k} \|d_{i,k}\|^2 \leq \tilde{\Delta}_{i,k}^* + \tilde{Q}_{i,k} & \text{when } \tilde{\Delta}_{i,k}^* > 0, \ \tilde{Q}_{i,k} > 0, \\ 0 \leq \Psi_{i,k} \|d_{i,k}\|^2 \leq \tilde{\Delta}_{i,k}^* & \text{when } \tilde{\Delta}_{i,k}^* > 0, \end{cases}$$

otherwise, if $i \in \mathscr{B}$:

$$\begin{cases} 0 \leq \Psi_{i,k} \|\hat{d}_{i,k}\|^2 \leq \tilde{\Delta}_{i,k}^* + \tilde{Q}_{i,k} & \text{when } \tilde{\Delta}_{i,k}^* > 0, \ \tilde{Q}_{i,k} > 0, \\ 0 \leq \Psi_{i,k} \|\hat{d}_{i,k}\|^2 \leq \tilde{\Delta}_{i,k}^* & \text{when } \tilde{\Delta}_{i,k}^* > 0. \end{cases}$$

Here, $\tilde{Q}_{i,k}$ is evaluated from (21) using (33) and, $\tilde{\Delta}_{i,k}^*$ from (32) using (34).

The following convergence theorem can now be established.

Theorem 2: Under Assumptions 1-3, and 5, the step size selection in (34) guarantees the convergence conditions stated in (6)-(8): if $i \in \mathscr{N}$, then $\lim_{k \to \infty} d_{i,k} = 0$, and, if $i \in \mathscr{B}$, then $\lim_{k \to \infty} \hat{d}_{i,k} = 0$.

Proof: The proof uses the same steps as in that of Theorem 1. The only difference lies in the use of new terms for $\tilde{\Delta}_{i,k}$, $\tilde{\Delta}_{i,k}^*$ and $Q_{i,k}$, given by (32), (34) and (33). Then, the final step of the proof is

$$\sum_{k=1}^{k_2'} \Psi_{i,k} [1_{\{i \in \mathscr{N}\}} \|d_{i,k}\|^2 + 1_{\{i \in \mathscr{B}\}} \|\hat{d}_{i,k}\|^2] \leq |\bar{B}_i| H_{UB}. \tag{36}$$

By Assumption 1, the R.H.S. of the above expression is finite and positive. Taking limits when $k_2' \to \infty$ yields the convergence criteria given in (7) and (8). Further, noting that Theorem 2 is a generalization of Theorem 1 with the step size selection scheme (34) replacing (18), it follows that (6) is also satisfied. ∎

C. Discussion

1) Extensions to time-varying graphs: Both the main problem (1) and the variable step size method (34) assume that agents are interconnected (i.e., inter agent communications occur) according to a fixed graph topology $\mathscr{G}$. As pointed out earlier, the nature of $\mathscr{G}$ can affect Assumption 3 (specifically through the T_i value) of the convergence proof. Nevertheless, due to the nature of the convergence proof we have used, it is reasonable to expect that the variable step size method (i.e., Theorem 2) is extendable to cases where the graph $\mathscr{G}$ varies sufficiently slower than the convergence rate and when it asymptotically converges to a fixed graph configuration. In fact, the class of coverage control problems which will be used to demonstrate the proposed DBS in Section IV belongs to the latter case. Moreover, since the variable step sizes in (34) lead each agent to maximize the improvement of its neighborhood objective function, we can expect convergence even when the graph $\mathscr{G}$ varies rapidly (however without showing any oscillatory behavior).

2) Feasible space constraint: When the main problem in (1) includes a feasible space constraint $\mathbf{s} \in \mathbf{F} \subset \mathbb{R}^{mN}$, we use the standard gradient projections [31] for (3) and (5). For such a situation, the following lemma presents an additional condition which needs to be satisfied in order to guarantee the convergence of the proposed variable step size method (34).

Lemma 6: If feasible space $\mathbf{F}$ is convex, and if an agent i's local and cross gradients satisfy the conditions,

$$\begin{aligned} |d_{i,k}^T \sum_{j\in B_i} d_{ij,k}| &< \|d_{i,k}\|^2 \text{ when } i \in \mathscr{N}, \\ |\hat{d}_{i,k}^T (\sum_{j\in B_i} d_{ij,k} + (d_{i,k} - \hat{d}_{i,k}))| &< \|\hat{d}_{i,k}\|^2 \text{ when } i \in \mathscr{B}, \end{aligned} \tag{37}$$

the step sizes $\beta_{i,k} = \beta_{i,k}^*$ given by (34) when used in (3) or (5) with standard gradient projections (onto $\mathbf{F}$), will lead the state $s_{i,k}$ to a stationary point (i.e., convergence).

Proof: Consider the problem where the neighborhood objective function $\tilde{H}_i(\tilde{s}_{i,k})$ is maximized using the projected state updates of $s_{i,k}$ on the convex feasible space $\mathbf{F}$. Following [31] in this situation, the convergence condition on the step sizes $\beta_{i,k}$ is $0 < \beta_{i,k} < \frac{2}{K_i}$, where K_i is the Lipschitz constant of $\nabla\tilde{H}_i$. Note that we can write $K_i = \sum_{j\in\bar{B}_i} K_{1j}$ due to (13). Also, for $i \in \mathscr{N}$, the expression for $\beta_{i,k}^*$ given in (34) can be modified into the form

$$\beta_{i,k}^* = \frac{1}{K_i}\left[1 + \frac{d_{i,k}^T \sum_{j\in B_i} d_{ij,k}}{\|d_{i,k}\|^2}\right]. \tag{38}$$

Now, enforcing the convergence condition $0 < \beta_{i,k}^* < \frac{2}{K_i}$ yields the first condition in (37). Similarly the second condition in (37) can be obtained when the expression for $\beta_{i,k}^*$, $i \in \mathscr{B}$, in (34) is considered. ■

From a practical standpoint, if the conditions in Lemma 6 are being violated during the gradient ascent process, the neighborhood reduction and/or weight factor manipulation techniques mentioned in Remark 2 can be used to change B_i and/or $\tilde{H}_i$ respectively so that these conditions are satisfied. We also note that knowledge of the feasible space constraint $\mathbf{s} \in \mathbf{F}$ in (1) can play an important role in designing boosting functions $\hat{d}_i = f_i(d_i, \mathbf{F})$, as further discussed in Section IV.

3) Variable step sizes compared to fixed step sizes: In a centralized setting, using a fixed step size for the gradient ascent is typically computationally inexpensive, and, if properly executed, can deliver a higher convergence rate compared to variable step size methods. However, in a distributed setting where agents independently and asynchronously alter the gradient direction ((3) and (5)), using a fixed step size (typically $\beta_{i,k} = \frac{1}{K_i}$) might not lead to good overall convergence properties. Further, establishing convergence in this case generally requires additional restrictive assumptions. In contrast, the proposed variable step size method has the following advantages: (i) It is designed so as to account for the distributed and cooperative nature of the underlying problem, (ii) Its convergence has been established by making only a few locally verifiable assumptions, (iii) It is not computationally heavy compared to line search methods, and, (iv) During different modes (boosting/normal) the step sizes are automatically adjusted. As a result, the variable step size method in applications has shown better convergence results compared to fixed step size methods (see Sections III-D and IV).

4) Termination conditions for modes: In practice, the equilibrium conditions $d_i = 0$ and $\hat{d}_i = 0$ used in boosting schemes should be replaced with appropriate termination conditions [31] such as $\|d_i\| \leq \varepsilon_1$ and $\|\hat{d}_i\| \leq \varepsilon_2$ (respectively) where $\varepsilon_1, \varepsilon_2$ are two preselected small positive scalars.

5) Escaping and converging to saddle points: Due to the non-convexity of the objective function, saddle points may exist in the feasible space. However, as shown in [25], [26], first-order methods (3) almost always avoid a large class of saddle points (called strict saddle points) inherently. Nevertheless, if boosting functions are deployed through (5), clearly, saddle points are easier to escape from compared to local minima. Moreover, even if the convergence criteria (6) - (8) lead to a saddle point, it will have a higher cost compared to initially attained local minima (or saddle points) as a result of the comparison stage used in boosting schemes (e.g., see "$H^B > H$" block in Fig. 1).

D. An example for the variable step size method

In this section, a simple example is provided to illustrate the operation and convergence (i.e., validity) of the proposed variable step size method. In this example, the local objective functions are restricted to take a quadratic form:

$$H_i(\bar{s}_i) = -\Big\|\sum_{j\in\bar{B}_i} A_{ij}s_j - b_i\Big\|_{C_i}^2 = -\|g_i(\bar{s}_i)\|_{C_i}^2, \tag{39}$$

where $A_{ij} \in \mathbb{R}^{r\times m}, b_i \in \mathbb{R}^r$ and $C_i \in \mathbb{R}^{r\times r}$ for any $i \in \mathscr{V}, j \in \bar{B}_i$. The weight matrix C_i is symmetric and positive definite. The weighted norm is defined as $\|v\|_C^2 = v^T C v$ with $v \in \mathbb{R}^r$ and $C \in \mathbb{R}^{r\times r}$. The parameter r represents the dimension of the local cost function. Assuming the parameters A_{ij}, b_i, C_i, $\forall i \in \mathscr{V}$, $\forall j \in \bar{B}_i$ and the graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ are predefined (for a given N, m and r value combination), the optimization problem we consider is

$$\mathbf{s}^* = [s_1^*, s_2^*, \ldots, s_N^*] = \arg\max_{\mathbf{s}} H(\mathbf{s}) = \sum_{i=1}^{N} H_i(\bar{s}_i). \tag{40}$$

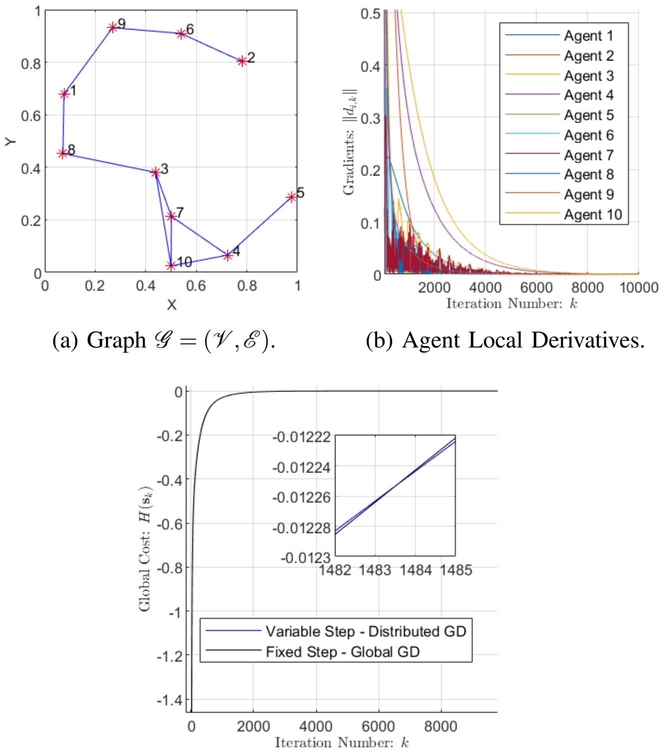
Due to the quadratic nature of the associated objective functions, a closed form expression can be obtained for the global optimum $\mathbf{s}^*$. Moreover, as a result of convexity, there is no need for any boosting function to escape an equilibrium point. Therefore, we use this example to compare the performance of the proposed variable step size method (when used in a distributed gradient ascent) to that of a fixed step size method (when used in a centralized gradient ascent).

For the (distributed) variable step size computation (at agent i using (18)), the local gradient d_i is

$$d_i = \frac{\partial H_i(\bar{s}_i)}{\partial s_i} = -2A_{ii}^T C_i g_i(\bar{s}_i), \tag{41}$$

the cross gradients d_{ij}, $\forall j \in \bar{B}_i$, are

$$d_{ij} = \left[\frac{\partial H_j(\bar{s}_j)}{\partial s_i}\right]_{i\leftrightarrow j} = -2A_{ji}^T C_j (\sum_{l\in\bar{B}_j} A_{jl}s_l - b_j), \tag{42}$$



(a) Graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$.

(b) Agent Local Derivatives.

(c) Global Objective Value.

Fig. 3: Numerical Example.

and, the local Lipschitz constants $K_{1j}, j \in \bar{B}_i$,

$$K_{1j} = 2\|A_j^T C_j A_j\|_\infty,\ A_j = [\{A_{jl}\}_{l\in\bar{B}_j}] \in \mathbb{R}^{r\times m|\bar{B}_j|}, \qquad (43)$$

are used. In contrast, in centralized gradient ascent, the global gradient component of agent i:

$$d_i^G = \frac{\partial H(\mathbf{s})}{\partial s_i} = -2\sum_{j\in\bar{B}_i} A_{ji}^T C_j g_j(\bar{s}_j), \qquad (44)$$

is used as a replacement for $d_{i,k}$ in (3). In this case, the step size is kept fixed at $\frac{1}{K_i}$ where $K_i = \sum_{j\in\bar{B}_i} K_{1j}$ (see [31]).

In simulations, fixed dimensional parameters $N = 10$ and $m = r = 2$ are used. Note that $m = r$ is required here to guarantee the existence of a solution where $d_i = 0, \forall i \in \mathscr{V}$. It it is easy to show that the optimal global objective function value is $H(\mathbf{s}^*) = 0$. To generate the inter-agent connections (i.e., the graph $\mathscr{G}$) a random geometric graph generation is used taking 0.4 as the communication range parameter [11]. The remaining problem parameters $A_{ij}, b_i, C_i, s_{i,0}\ \forall i \in \mathscr{V},\ \forall j \in \bar{B}_i$ are generated randomly (keeping the graph $\mathscr{G}$ fixed).

The experimental results shown in Fig. 3 confirm our theoretical convergence results. The $H(\mathbf{s}_k)$ profiles in Fig. 3(c) show that the proposed distributed variable step size method provides a slightly faster convergence than the centralized fixed step size method for $k \leq 1483$ where at $k = 1483$, the $H(\mathbf{s}_k)$ value is 99.95% closer to the optimal than the initial value $H(\mathbf{s}_0) = 26.1432$; for $k \geq 1484$, the centralized fixed step method is slightly faster. This cross-over behavior can be understood as a result of local gradients $d_{i,k}$ becoming smaller as k increases and adapting step sizes $\beta_{i,k}$ in (3) when $d_{i,k}$ is very small is less effective. Our general observation over extensive similar examples is that the result of such a comparison (between distributed variable step and centralized fixed step methods) depends on the network topology (additional results are provided in [28]).

IV. Application to Coverage Control Problems

In this section, we apply the theory developed in the previous sections to construct a convergence-guaranteed DBS based on two new boosting function families created for the class of multi-agent coverage control problems. As mentioned in the Introduction, the work in [12] has extended the solution proposed in [1] for this class of problems by adding the capability to escape local optima through a centralized boosting scheme (albeit, without any convergence guarantees). In contrast, we will develop a distributed gradient ascent based scheme which is guaranteed to converge based on Theorem 2 and demonstrate the use of boosting functions designed for this problem setting.

The coverage control problem aims to find an optimal arrangement for a given set of sensor nodes (agents) inside a given *mission space* so as to maximize the probability of detecting randomly occurring events. It is assumed that the agent sensing capabilities, characteristics of the mission space, and any a priori information on the spacial likelihood of random event occurrences (in the mission space) are fixed and known.

The mission space $\Omega \subset \mathbb{R}^2$ is modeled as a non-self-intersecting polygon [1], and it may contain a finite set of non-self-intersecting polygonal obstacles denoted by $\{M_1, M_2, \dots, M_h\}$, where $M_i \subset \mathbb{R}^2$ represents the interior space of the i^{th} obstacle. Therefore, agent motion and deployment are constrained to a non-convex feasible space $F = \Omega \backslash (\cup_{i=1}^h M_i)$. Note that "$\backslash$"denotes the set subtraction operator. The spacial likelihood of random event occurrence over the mission space is quantified by the *event density* function $R : \Omega \to \mathbb{R}$, where, $R(x) \geq 0, \forall x \in \Omega$; $R(x) = 0,\ \forall x \notin F$, and $\int_\Omega R(x)dx < \infty$ are assumed. If no a priori information related to $R(x)$ is available, then $R(x) = 1, \forall x \in \Omega$ is used.

The mission space is considered to have N agents. Following the same notation used in Section II and III, at a given discrete update instant k, the position of agent i (i.e., the controllable *local state*) is denoted by $s_{i,k} \in F \subset \mathbb{R}^2$ and the *global state* of the multi-agent system is $\mathbf{s}_k = [s_{1,k}, s_{2,k}, \dots, s_{N,k}] \in \mathbb{R}^{2N}$. We write $\mathbf{s}_k \in \mathbf{F}$ to denote $s_{i,k} \in F\ \forall i$. For notational convenience, the update instant index k is omitted unless it is important.

The sensing capabilities of agent i depend on: (i) A finite *sensing radius* $\delta_i \in \mathbb{R}$ beyond which it cannot detect any events, (ii) The presence of obstacles which hinder its sensing capability. Considering these two factors, a *visibility region* for agent i is defined as $V_i = \{x : \|x - s_i\| \leq \delta_i, \forall \lambda \in (0,1], (\lambda x + (1-\lambda)s_i) \in F\}$. Fig. 4 is provided to identify all associated geometric parameters in this model.

A *sensing function* $\hat{p}_i(x, s_i)$ is used to quantify the probability that "agent i detects an event occurring at $x \in F$." Due to the physical limitations mentioned above, $\hat{p}_i(x, s_i) = 1_{\{x\in V_i\}} p_i(x, s_i)$ where $1_{\{\cdot\}}$ is the usual indicator function and

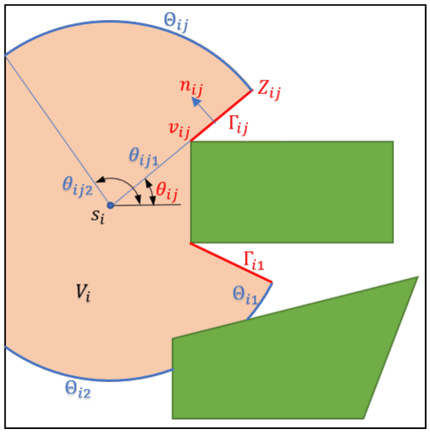


Fig. 4: Mission space with one agent.

$p_i(x,s_i)$ is defined so that $p_i:\mathbb{R}^2\times\mathbb{R}^2\to\mathbb{R}$ and is differentiable and monotonically decreasing in $D_i(x)\equiv\|x-s_i\|$. For example, $p_i(x,s_i)=p_{0i}e^{-\lambda_i D_i(x)}$ is a typical choice. However, note that $\hat{p}_i(x,s_i)$ is strictly discontinuous w.r.t. x, s_i or $D_i(x)$. Assuming independently detecting agents, the *joint detection probability* $P(x,\mathbf{s})$, i.e., the probability of "detecting an event occurring at $x\in F$ by at least one agent," is given by

$$P(x,\mathbf{s})=1-\prod_{i=1}^{N}[1-\hat{p}_i(x,s_i)]. \tag{45}$$

Combining the event density and joint detection probability, the objective function $H(\mathbf{s})$ of the coverage control problem given in [1] is

$$H(\mathbf{s})=\int_F R(x)P(x,\mathbf{s})dx, \tag{46}$$

and the multi-agent optimization problem is

$$\mathbf{s}^*=\arg\max_{\mathbf{s}\in\mathbf{F}} H(\mathbf{s}). \tag{47}$$

where $\mathbf{s}^*$ represents the optimal agent placement. Note that the objective function in (46) is non-linear and non-convex, while the feasible space $\mathbf{F}$ is also non-convex. Therefore, the coverage control problem posed in (47) has the same structure as the general cooperative multi-agent optimization problem in (1). Thus, (47) can have multiple locally optimal solutions (even in the simplest configurations). Therefore, the use of the DBS with appropriate boosting functions can aid the agents to escape local optima while solving (47).

A. Distributed optimization solution

If two agents have an overlap in their visibility regions, they are considered as *neighbors* [12]. Therefore, the *neighborhood* B_i and the *closed neighborhood* $\bar{B}_i$ of an agent i are the sets defined as $B_i=\{j:V_j\cap V_i\neq\emptyset, i\neq j\}$ and $\bar{B}_i=B_i\cup\{i\}$ respectively. It is assumed that agents share their local state information s_i with their neighbors, so that each agent has knowledge of its *neighborhood state* $\bar{s}_i\equiv\{s_j:j\in\bar{B}_i\}$. We use an undirected graph $\mathscr{G}=(\mathscr{V},\mathscr{A})$ to model inter-agent interactions, where $\mathscr{V}=\{1,2,\ldots,N\}$ and $\mathscr{A}=\{(i,j):i,j\in\mathscr{V},\ i\neq j,\ j\in B_i\}$.

In [12], it is shown that the coverage global objective $H(\mathbf{s})$ in (46) can be expressed as $H(\mathbf{s})=H_i(\bar{s}_i)+H_i^c(s_i^c)$, where

$$H_i(\bar{s}_i)=\int_{V_i}R(x)\prod_{j\in B_i}[1-\hat{p}_j(x,s_j)]\,p_i(x,s_i)dx, \tag{48}$$

and $H_i^c(s_i^c)=\int_F R(x)(1-\prod_{j\in\mathscr{V}-\{i\}}[1-\hat{p}_j(x,s_j)])dx$ with $s_i^c=\{s_j:j\in\mathscr{V}-\{i\}\}$. Thus, the $H_i(\bar{s}_i)$ term only depends on the neighborhood state and is called the *local objective* function, while $H_i^c(s_i^c)$ is independent of s_i. As a result of this property, the local gradient of agent i, defined as $d_i=\frac{\partial H_i(\bar{s}_i)}{\partial s_i}\in\mathbb{R}^2$, is always equal to the global gradient component $\frac{\partial H(\mathbf{s})}{\partial s_i}$. Therefore, each agent i can evaluate its global gradient component using only its own local objective function $H_i(\cdot)$ and the neighborhood state $\bar{s}_i$ and the distributed gradient ascent scheme in (3) (i.e., $s_{i,k+1}=s_{i,k}+\beta_{i,k}d_{i,k}$) can be used to solve the problem in (47) in a distributed manner. In order to execute (3), each agent must evaluate its local gradient $d_{i,k}$ and select its step size $\beta_{i,k}$ appropriately. Section IV-A1 provides the derivation of $d_{i,k}$ and analyzes its structure which is pivotal in effectively designing boosting functions.

1) Derivation of the gradient $d_{i,k}$: Observing that the gradient d_i is a two-dimensional vector, we write $d_i=[d_{iX},d_{iY}]^T$ and use the Leibniz rule [32] in (46) to express d_{iX} as

$$\begin{aligned} d_{iX}=\frac{\partial H_i(\bar{s}_i)}{\partial s_{iX}}&=\int_{V_i}R(x)\Phi_i(x)\frac{\partial p_i(x,s_i)}{\partial s_{iX}}dx\\ &\quad+\int_{\partial V_i}R(x)\Phi_i(x)p_i(x,s_i)\underline{V_x}\cdot\underline{n_x}dl, \end{aligned} \tag{49}$$

where,

$$\Phi_i(x)=\prod_{j\in B_i}[1-\hat{p}_j(x,s_j)]. \tag{50}$$

The second term in (49) is a line integral over the boundary of the sensing region ∂V_i. The terms $\underline{V_x}$ and $\underline{n_x}$ respectively represents the rate of change and the unit normal vector of ∂V_i at x due to an infinitesimal change in s_{iX}, where $s_i=[s_{iX},s_{iY}]^T$.

From Fig. 4, notice that the shape of a boundary ∂V_i is formed by: (i) mission space boundaries, (ii) obstacle edges, (iii) obstacle vertices, and, (iv) sensing range. However, when s_{iX} (or s_{iY}) is perturbed infinitesimally, $\underline{V_x}\neq 0$ only when x lies on ∂V_i components formed due to the latter two factors. Therefore, we label the linear segments of ∂V_i formed due to obstacle vertices as $\Gamma_i=\{\Gamma_{i1},\Gamma_{i2},\ldots\}$ and the circularly shaped curves formed due to a finite sensing range as $\Theta_i=\{\Theta_{i1},\Theta_{i2},\ldots\}$.

The first term in (49) can be simplified using the relationship between the sensing function $p_i(x,s_i)$ and $D_i(x)$. Further, considering the behavior of $\underline{V_x}\cdot\underline{n_x}$ on the segments in Γ_i and Θ_i sets, the line integral part of (49) can also be simplified to get two additional terms. The resulting complete expression for d_{iX} is

$$\begin{aligned} d_{iX}&=\int_{V_i}w_{i1}(x,\bar{s}_i)\frac{(x-s_i)_X}{\|x-s_i\|}dx\\ &\quad+\sum_{\Gamma_{ij}\in\Gamma_i}sgn(n_{ijX})\frac{\sin\theta_{ij}}{\|v_{ij}-s_i\|}\int_0^{Z_{ij}}w_{i2}(\rho_{ir}(r),\bar{s}_i)rdr\\ &\quad+\sum_{\Theta_{ij}\in\Theta_i}\delta_i\cos\theta\int_{\theta_{ij1}}^{\theta_{ij2}}w_{i3}(\rho_{i\theta}(\theta),\bar{s}_i)d\theta, \end{aligned} \tag{51}$$

where, $sgn(\cdot)$ is the signum function, and we define:

$$w_{i1}(x,\bar{s}_i) = -R(x)\Phi_i(x)\frac{dp_i(x,s_i)}{dD_i(x)},$$
$$w_{i2}(x,\bar{s}_i) = w_{i3}(x,\bar{s}_i) = R(x)\Phi_i(x)p_i(x,s_i),$$

with $\rho_{ir}(r) = \frac{v_{ij}-s_i}{||v_{ij}-s_i||}r + v_{ij}$ and $\rho_{i\theta}(\theta) = s_i + \delta_i[\cos\theta\ \sin\theta]^T$.

As seen in Fig. 4, a line segment $\Gamma_{ij} \in \Gamma_i$ is characterized by its geometric parameters: end point Z_{ij}, angle θ_{ij}, obstacle vertex v_{ij}, and direction $n_{ij} = [n_{ijX}, n_{ijY}]^T$. Thus, each Γ_{ij} is a 4-tuple $(Z_{ij}, \theta_{ij}, v_{ij}, n_{ij})$. Similarly, a circular arc segment $\Theta_{ij} \in \Theta_i$ is quantified by starting angle θ_{ij1} and ending angle θ_{ij2}. Therefore, each Θ_{ij} term is 2-tuple $(\theta_{ij1}, \theta_{ij2})$.

The complete expression in (51) can be thought of as a sum of forces acting on agent i, generated by different points $x \in V_i$. In (51), the weight function $w_{i1}(x,\bar{s}_i)$ represents the magnitude of the force pulling agent i towards point $x \in V_i$. Similarly, $w_{i2}(x,\bar{s}_i)$ describes the force generated by a point $x \in \Gamma_{ij}$ in the lateral direction to the line Γ_{ij} (inwards the region V_i). Finally, $w_{i3}(x,\bar{s}_i)$ represents the magnitude of the attraction force generated by (and towards) a point $x \in \Theta_{ij}$. From this interpretation, the gradient component d_{iX} can be viewed as a function of three weight functions: $d_{iX} = d_{iX}(w_{i1}, w_{i2}, w_{i3})$. This representation is instrumental for the construction of boosting functions.

The same procedure can be followed in deriving d_{iY} (details in [28]). Therefore, each agent can execute (3) as it can locally evaluate its normal gradient $d_{i,k}$ at each update instant k.

B. Designing boosting functions

We now focus on constructing an appropriate expression for the boosted gradient $\hat{d}_{i,k}$ to be used in (5) for the class of coverage control problems. In deriving (51), the relationship $d_i = d_i(w_{i1}, w_{i2}, w_{i3})$ was identified where each weight function $w_{ij} = w_{ij}(x,\bar{s}_i)$ represents the magnitude component of each of three infinitesimal forces, $j = 1,2,3$, acting on agent i generated at a point $x \in V_i$. Note that $d_{i,k} = 0$ only occurs when all the aforementioned infinitesimal forces add up to a resultant force with zero magnitude. Therefore, by appropriately transforming the weight functions $\{w_{ij}(x,\bar{s}_i) : j = 1,2,3\}$, a valid expression for $\hat{d}_{i,k}$ can be constructed which avoids such equilibrium configurations. Specifically, we consider weight function transformations given by

$$\hat{w}_{ij}(x,\bar{s}_i) = \alpha_{ij}(x,\bar{s}_i)w_{ij}(x,\bar{s}_i) + \eta_{ij}(x,\bar{s}_i), \quad j = 1,2,3. \tag{52}$$

Here, both $\alpha_{ij}, \eta_{ij} : \mathbb{R}^2 \times \mathbb{R}^{2|\bar{B}_i|} \to \mathbb{R}$ are known as *transformation functions*. The resulting boosted gradient $\hat{d}_{i,k}$ expression takes the form

$$\hat{d}_{i,k} = d_{i,k}(\hat{w}_{i1}, \hat{w}_{i2}, \hat{w}_{i3}). \tag{53}$$

Compared to heuristic methods where the gradient is randomly perturbed (to escape local optima), the use of boosted gradient direction $\hat{d}_{i,k}$ given in (53) is a far more "intelligent" choice as long as each agent i chooses its transformation functions α_{ij}, η_{ij}, $j = 1,2,3$, so as to trigger a systematic exploration of the mission space. This is discussed next.

1) Boosting function families: A boosting function family is characterized by the form of the transformation functions $\alpha_{ij}(x,\bar{s}_i)$, $\eta_{ij}(x,\bar{s}_i)$, $j = 1,2,3$, in (52). As a result, different boosting function families exhibit different properties. Here, we briefly review three boosting function families proposed in [12], and introduce two new ones.

The underlying rationale behind constructing a boosting function family lies in answering the question: "Once an agent converges under the normal gradient-based mode, how can the agent escape the current equilibrium towards a direction giving a high priority to points likely to achieve a higher objective function value?" Towards this goal, to define appropriate $\alpha_{ij}(x,\bar{s}_i)$, $\eta_{ij}(x,\bar{s}_i)$, $j = 1,2,3$, in (52), the information available to agent i consists of: (i) The neighborhood state $\bar{s}_i$, (ii) The local objective function $H_i(\cdot)$, (iii) The neighboring mission space topological information contained in Γ_i and Θ_i (see Fig. 4), (iv) Past state trajectory information $\{s_{i,k} : k < k_1\}$. The three boosting function families proposed in [12] use $\bar{s}_i$ and $H_i(\cdot)$, whereas the two new ones make use of Γ_i, Θ_i and $\{s_{i,k} : k < k_1\}$ in addition to the information of $\bar{s}_i$ and $H_i(\cdot)$.

In the following discussion, for notational convenience, we refer to the setting where $\alpha_{ij}(x,\bar{s}_i) = 1$, $\eta_{ij}(x,\bar{s}_i) = 0$, $j = 1,2,3$, as the *default configuration* in (52). Also, note that κ and γ used in defining boosting function families always act as two positive *gain parameters*.

***Φ-Boosting** [12]:* This method uses $\alpha_{i1}(x,\bar{s}_i) = \kappa\Phi_i(x)^\gamma$ and $\eta_{i1}(x,\bar{s}_i) = 0$, where $\Phi_i(x)$ in (50) indicates the extent to which point $x \in V_i$ is *not* covered by neighbors in B_i. Thus, the effect of Φ-Boosting is to force agent i to move towards regions of V_i which are less covered by its neighbors in B_i.

***P-Boosting** [12]:* In this method, $\alpha_{i1}(x,\bar{s}_i) = \kappa[P(x,\mathbf{s})]^{-\gamma}$ and $\eta_{i1}(x,\bar{s}_i) = 0$ are used, where $P(x,\mathbf{s})$ in (45) indicates the extent to which point $x \in \Omega$ is covered by all the agents in $\mathscr{V}$. However, when evaluating the boosted gradient (53), $x \in V_i \subseteq \Omega$. Therefore, this approach assigns higher weights to points $x \in V_i$ which are less covered by the closed neighborhood $\bar{B}_i$.

***Neighbor-Boosting** [12]:* This boosting function family uses $\alpha_{i1}(x,\bar{s}_i) = 1$ and $\eta_{i1}(x,\bar{s}_i) = \sum_{j\in B_i} 1_{\{x=s_j\}} \cdot \frac{\kappa \cdot 1_{\{s_j \in V_i\}}}{\|s_i - x\|^\gamma}$. As a result, agent i gets repelled from the neighbors who are in its visibility region V_i.

Note that these boosting methods are limited to transforming the first integral term of the gradient expression in (51), i.e., only the weight $w_{i1}(x,\bar{s}_i)$ through $\alpha_{i1}(x,\bar{s}_i)$, $\eta_{i1}(x,\bar{s}_i)$ is transformed. Next, we present two new boosting function families.

***V-Boosting**:* The V-Boosting function uses the information of obstacle vertices $v_{ij} \in \Gamma_{ij} \in \Gamma_i$ which lie inside V_i so as to navigate an agent i around surrounding obstacles. This method is inspired by the second integral term in (51) which represents the effect of obstacles through Γ_i in V_i on agent i. Therefore, in V-Boosting, the weight function $w_{i2}(x,\bar{s}_i)$ is transformed via the $\eta_{i2}(x,\bar{s}_i)$ term so that the second integral term in (51) is modified. Specifically,

$$\alpha_{i1}(x,\bar{s}_i) = \kappa_1\Phi_i(x)^{\gamma_1}(1 - p_i(x,s_i)), \tag{54}$$
$$\eta_{i2}(x,\bar{s}_i) = 1_{\{x=Z_{ij}\}} \cdot \kappa_2\|x - s_i\|^{\gamma_2}. \tag{55}$$

Moreover, note that $w_{i1}(x,\bar{s}_i)$ is also transformed via the $\alpha_{i1}(x,\bar{s}_i)$ term as in both Φ-Boosting and P-Boosting. In all, the transformation in (54) forces agent i to move toward less covered areas while the transformation in (55) acts as an attraction force directed towards $Z_{ij} \in \Gamma_{ij}$ (same as in the direction of obstacle vertex v_{ij}). The combination of these two influences enables agent i to navigate around obstacles aiming to expand the mission space exploration.

Arc-Boosting: The Arc-Boosting method uses the information in Θ_i to transform the weight function $w_{i3}(x,\bar{s}_i)$. This involves the third term in (51) which was not previously included in prior work [1], [12]. Note that $\{\theta_{ij1},\theta_{ij2}\} = \Theta_{ij} \in \Theta_i$ represents a circular *arc* formed due to the finite sensing range and obstacles. Based on the an agent's location in the mission space relative to the surrounding obstacles, it can have multiple arcs in its boundary ∂V_i. For example, the agent in Fig. 4 has three such arcs. Under the Arc-Boosting method, first, each arc segment $\Theta_{ij} \in \Theta_i$ is classified into one of three disjoint sets: (i) Attractive Arcs Θ_i^+, (ii) Repulsive Arcs Θ_i^-, and (iii) Neutral Arcs Θ_i^0. This classification is based on the metric $A(\Theta_{ij})$:

$$A(\Theta_{ij}) = \frac{1}{(\theta_{ij2}-\theta_{ij1})}\int_{\theta_{ij1}}^{\theta_{ij2}}(1-\prod_{k\in\bar{B}_i}(1-\hat{p}_k(\rho_{i\theta}(\theta),s_k)))d\theta,$$

which measures the mean coverage level on the arc segment Θ_{ij} by the agents in the closed neighborhood $\bar{B}_i$. Specifically, the arc with the maximum $A(\Theta_{ij})$ value is assigned to be a repulsive arc (i.e., in the set Θ_i^+), while the arc with the minimum $A(\Theta_{ij})$ value is assigned to be an attractive arc (i.e., in the set Θ_i^-). The remaining arcs are labeled as neutral (i.e., in the set Θ_i^0). However, it is possible that an equilibrium occurs (i.e., $A(\Theta_{ij})$ are identical for all j), which may happen when $B_i = \emptyset$. In this case, we use a recent state $s_{i,k-K}$, where $K \geq 1$ as a parameter of the Arc-Boosting method, selected from the agent's own past state trajectory. Specifically, the arc which is in the direction of $s_{i,k-K}$ (from point s_i) is regarded as a repulsive arc while all other arcs are labeled as attractive.

Based on the arc partition given by Θ_i^+, Θ_i^- and Θ_i^0, the Arc-Boosting function family is formally defined by the weight function $w_{i3}(x,\bar{s}_i)$ transformation given by

$$\alpha_{i3}(x,\bar{s}_i) = 1_{\{\Theta_{ij}\in\Theta_i^0\}}, \tag{56}$$

$$\eta_{i3}(x,\bar{s}_i) = [1_{\{\Theta_{ij}\in\Theta_i^+\}} - 1_{\{\Theta_{ij}\in\Theta_i^-\}}]\cdot F_c(\kappa,\gamma). \tag{57}$$

In (57), the value of the term in brackets is either $1,-1$ or 0 depending on whether Θ_{ij} belongs to Θ_i^+, Θ_i^- or Θ_i^0 respectively. The term $F_c(\kappa,\gamma)$ is a gain factor where a typical choice would be of the form $F_c(\kappa,\gamma) = \kappa e^{\gamma}$.

The motivation behind the Arc-Boosting method is to encourage agent i to: (i) Move away from highly covered regions (i.e., from repulsive arcs), (ii) Move towards less covered regions (i.e., towards attractive arcs), and, (iii) Move continuously towards unexplored regions (i.e., towards an opposing direction to the already visited point $s_{i,k-K}$). As will be seen in Section IV-D, the Arc-Boosting family has been found to be the most effective in handling the presence of multiple obstacles/constraints within V_i.

TABLE I: Boosting function parameters used in simulations.

Boosting Method	Associated Default Parameters
P-Boosting	$\kappa = 1,\ \gamma = 1$
Neighbor-Boosting	$\kappa = 10000,\ \gamma = 1$
Φ-Boosting	$\kappa = 4,\ \gamma = 2$
V-Boosting	$\kappa_1 = 10,\ \kappa_2 = 5,\ \gamma_1 = 1$, and, $\gamma_2 = 1$
Arc-Boosting	$\kappa = 1,\ \gamma = 1,\ K = 50,\ T_D = 5$

C. DBS for coverage control

We are now ready to apply the DBS in Fig. 2 to the class of coverage control problems (complete implementation details are given in [28] including some refinements of the scheme and improvements enabled by using an alternative function $\bar{H}_i(\bar{s}_i)$ compared to the one in (13)). Convergence is guaranteed through Theorem 2 by checking that Assumptions 1-3 and 5 are all satisfied. Assumption 1 holds for the coverage control problem due to two reasons: (i) The Lipshitz constant K_{1i} of $\nabla H_i(\bar{s}_i)$ can be locally evaluated and, as shown in [28], whenever the sensing capabilities are smooth (i.e. $p_i(x,s_i)$ is differentiable w.r.t. $D_i(x)$) the computed K_{1i} value are always finite. (ii) $H_{UB} = \int_{V_i} R(x)dx$ is a typical upper bound for $H_i(\bar{s}_i)$ as $\int_\Omega R(x)dx < \infty$ is already enforced in the coverage control problem formulation. Assumption 2 holds for coverage control problem because information sharing capability is already assumed in the basic coverage control problem framework [1], [12]. However, the following lemma is useful in asserting that no additional communication bandwidth is required to satisfy this assumption.

Lemma 7: For the class of coverage control problems, any agent $i \in \mathscr{V}$ can locally compute $d_{ij} = \frac{\partial H_j(\bar{s}_j)}{\partial s_i}$ value $\forall j \in \bar{B}_i$.

Proof: By taking the partial derivative of (48) (written for agent j) w.r.t. the local state s_i yields

$$d_{ij} = -\int_{V_j} R(x)p_j(x,s_j)\prod_{l\in B_j-\{i\}}(1-p_l(x,s_l))\frac{dp_i(x,s_i)}{ds_i}dx.$$

Now, note that $\forall x \notin V_i$, $\frac{-dp_i(x,s_i)}{ds_i} = 0$, and, $\forall l \notin B_i \cap B_j, \forall x \in V_i \cap V_j$, $p_l(x,s_l) = 0$. By incorporating these relationships into the obtained expression for d_{ij} gives a locally computable (at agent i) expression for d_{ij} as

$$d_{ij} = -\int_{V_i\cap V_j} R(x)p_j(x,s_j)\prod_{l\in B_i\cap B_j}(1-p_l(x,s_l))\frac{dp_i(x,s_i)}{ds_i}dx. \blacksquare$$

Assumption 3 has been previously justified for the general setting in Section III using Lemmas 3 and 5. However, to ensure this assumption is satisfied in coverage control problems, the parameter T_i was observed during all simulations presented in Section IV-D for all agents. In all occasions, T_i was found to be a finite value consistent with Assumption 3. One such observed T_i value distribution is given in Fig. 5, where 99.1% of the time $T_i \leq 10$. Finally, as pointed out in Section III, Assumption 5 is trivial and will hold for any general cooperative multi-agent problem including coverage control problems.

D. Simulation results

For the class of coverage control problems, the proposed DBS (with all boosting function families) and the

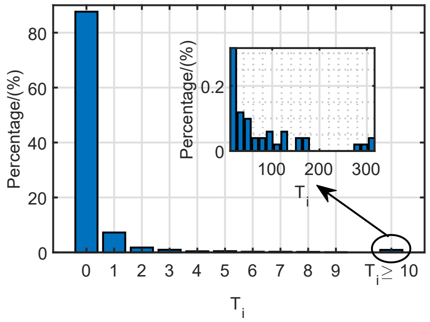
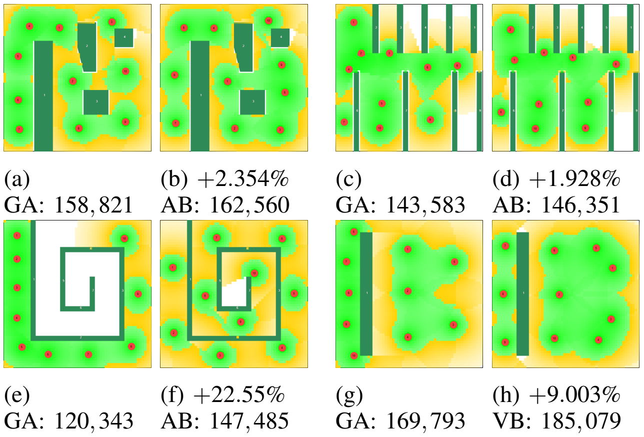


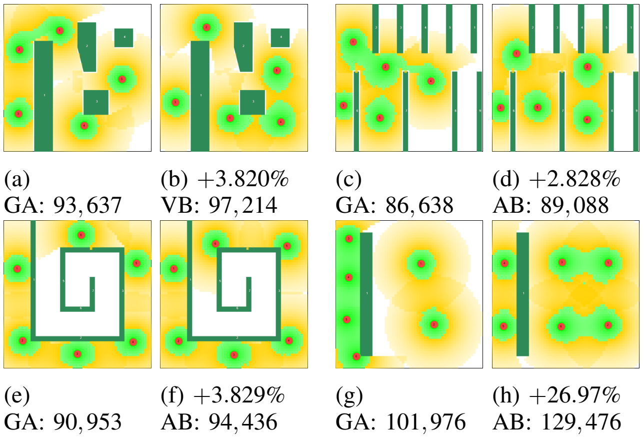
Fig. 5: Percentage occurrence of different T_i values (Regarding Assumption 3 for the simulation which produced the result shown in Fig. 6f).



(a) GA: $158,821$ (b) $+2.354\%$ AB: $162,560$ (c) GA: $143,583$ (d) $+1.928\%$ AB: $146,351$

(e) GA: $120,343$ (f) $+22.55\%$ AB: $147,485$ (g) GA: $169,793$ (h) $+9.003\%$ VB: $185,079$

Fig. 6: Maximum coverage improvement achieved due to boosting for $N = 10$ (See Tab. II).



(a) GA: $93,637$ (b) $+3.820\%$ VB: $97,214$ (c) GA: $86,638$ (d) $+2.828\%$ AB: $89,088$

(e) GA: $90,953$ (f) $+3.829\%$ AB: $94,436$ (g) GA: $101,976$ (h) $+26.97\%$ AB: $129,476$

Fig. 7: Maximum coverage improvement achieved due to boosting for $N = 5,6$ (See Tab. III).

methods proposed in [1], [12] were implemented in an interactive JavaScript-based simulator which is available at http://www.bu.edu/codes/simulations/shiran27/CoverageFinal/ and may be used by the reader to reproduce the reported results. The boosting function parameters used in generating the results reported next (i.e., the gain parameters κ, γ) are given in Table I.

Based on the obstacle arrangement, four different mission space configurations named ‘General’,‘Room’,‘Maze’, and ‘Narrow’ are considered in the simulations. In Figs. 6, 8, 9, and, 7 obstacles are shown as green-colored blocks and agent locations are shown in red-colored dots. In all experiments, agents have been initialized at the top left corner of the mission space. Further, light green-colored areas indicate higher coverage levels while yellow-colored areas indicate the opposite.

As the first step, a set of experiments was conducted with $N = 10$ agents and three different algorithms were tested: (i) The conventional distributed gradient ascent method proposed in [1] (labelled ”GA”), (ii) The centralized boosting scheme (CBS) proposed in [12], and, (iii) The distributed boosting scheme (DBS) proposed in this paper.

Results obtained from the GA method are shown in Figs. 6a, 6c, 6e, and, 6g. The corresponding objective function values are listed in Table II under the column: ‘Reference Level $H(\mathbf{s}^1)$’. Similarly, results obtained from the CBS and DBS methods (under different boosting function families) are listed in the remaining columns of Table II - as the increment achieved in the coverage objective value with respect to the reference level $H(\mathbf{s}^1)$. Also note that, as another baseline for the proposed boosting methods, a random gradient perturbation method is also implemented which uses $\hat{d}_{i,k} = d_{i,k} + \kappa\zeta_{i,k}$ during the boosting sessions. Here, $\kappa = 5$ and $\zeta_{i,k} \in \mathbb{R}^2$ is a two-dimensional random vector, independently generated from a standard uniform distribution at each time step.

The cases with the highest coverage objective value increments are shown in bold letters and they are illustrated in Figs. 6b, 6d, 6f and 6h. The results in Table II show that the distributed Arc-Boosting (labeled ”AB”) and distributed V-Boosting (labeled ”VB”) schemes outperform all other methods for all tested obstacle configurations when $N = 10$.

Moreover, to further investigate the performance of the distributed V-Boosting and Arc-Boosting methods, simulation results were generated with moderate N values such as $N = 5,6$. The corresponding results are shown in Table III and Fig. 7. For comparison purposes, some results obtained with the distributed Φ-Boosting (labeled “ΦB”, with $N = 5,6,10$) are also shown in Fig. 8. Finally, extreme situations (i.e., more prone to local optima) where very few agents are deployed (i.e., $N = 1,2$) were also investigated and simulation results obtained are shown in Table IV and Fig. 9. Note that these additional experimental results also highlight the impact of the distributed Arc-Boosting and V-Boosting schemes.

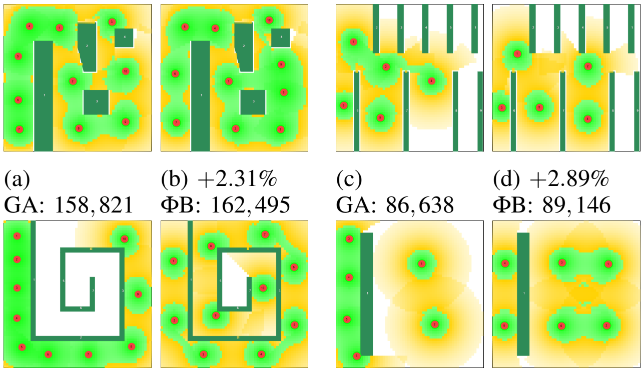
In summary, these simulation results show that the boosting function approach can successfully escape local optima which limit the conventional gradient ascent based method. Further, the systematic gradient transformation process achieved by the specifically designed boosting function families, along with the introduced DBS, delivers superior objective function values compared to conventional gradient ascent based methods, as well as compared to random gradient perturbation based techniques. Additional implementation details such as the selection of gain parameters in the boosting functions and terminal conditions for the algorithms used are provided in [28].

TABLE II: Coverage objective value **increment** (+/-) achieved by different boosting schemes (See Fig. 6).

		Reference Level $H(\mathbf{s}^1)$	**Coverage objective value increment occurred with respect to the 'Reference Level $H(\mathbf{s}^1)$'**											
Configuration		Gradient	Random Pert.		*P*-Boosting		Neighbor Boo.		Φ-Boosting (ΦB)		V-Boosting (VB)		Arc-Boosting (AB)	
Obstacles	N	Ascent (GA)	Centr.	Decen.	Centr.	Decen.	Centr.	Decen.	Centr.	Decen.	Centr.	Decen.	Centr.	Decen.
General	10	**158,821**	+233	+409	+235	+3684	+235	+3676	+243	+3674	+2453	+3621	+3553	**+3739**
Room	10	**143,583**	+1366	+484	+1578	+2680	+2374	+968	+1578	+2626	+1739	+2455	+1578	**+2768**
Maze	10	**120,343**	+20037	+19409	+25937	+25897	+19443	+25895	+26952	+23868	+19970	+25702	+25945	**+27142**
Narrow	10	**169,793**	+150	+8781	+9204	+8835	+15258	+9391	+15008	+9376	+14969	**+15286**	+15238	+15120

TABLE III: Coverage objective value for cases with $N = 5,6$ with decentralized boosting (See Fig. 7).

Configuration		Gradient	Decentralized	Decentralized
Obstacles	N	Ascent (GA)	V-Boosting	Arc-Boosting
General	5	**93,637**	**97,214**	96,832
Maze	6	**90,953**	94,026	**94,436**
Room	5	**86,638**	89,078	**89,088**
Narrow	6	**101,976**	116,481	**129,476**

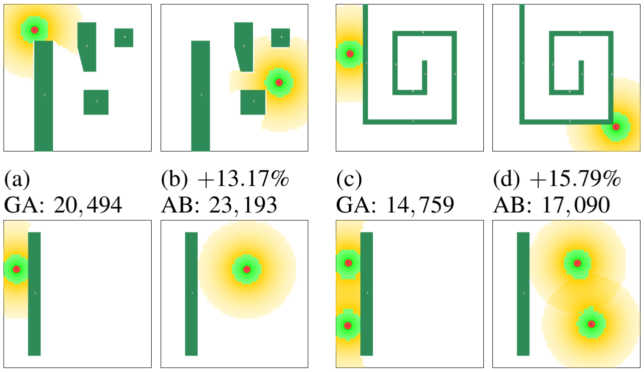


(a) GA: 158, 821 (b) +2.31% ΦB: 162, 495 (c) GA: 86, 638 (d) +2.89% ΦB: 89, 146

(e) GA: 120, 343 (f) +19.8% ΦB: 144, 211 (g) GA: 101, 976 (h) +27.0% ΦB: 129, 542

Fig. 8: Coverage improvement due to distributed Φ-Boosting for $N = 10, 5, 6$ situations.

It should be emphasized that whenever the DBS was used, the variable step size method involved in Theorem 2 was used to guarantee convergence. As an example, Fig. 10 illustrates the observed step size sequence and the associated gradient sequence of a typical agent ($i = 4$) during the simulation which leads to the result shown in 6h. Moreover, Table V provides a comparison of coverage objective and convergence time values observed for the DBS when fixed and variable step sizes are used. Note that the use of variable step sizes has improved (i.e., reduced) the convergence time by 61.9% (i.e., by 203.5s). These convergence times were observed on an Intel® Core$^{\text{TM}}$ i7-8700 CPU @3.20 GHz Processor with a 32 GB RAM.



(a) GA: 20, 494 (b) +13.17% AB: 23, 193 (c) GA: 14, 759 (d) +15.79% AB: 17, 090

(e) GA: 13, 669 (f) +121.4% VB: 30, 259 (g) GA: 26, 258 (h) +123.5% VB: 58, 693

Fig. 9: Maximum coverage improvement achieved due to boosting for $N = 1, 2$ (See Tab. IV).

TABLE IV: Coverage objective value for cases with $N = 1,2$ with decentralized boosting (See Fig. 9).

Configuration		Gradient	Decentralized	Decentralized
Obstacles	N	Ascent (GA)	V-Boosting	Arc-Boosting
General	1	**20,494**	20,404	**23,193**
Maze	1	**14,759**	14,774	**17,090**
Narrow	1	**13,669**	**30,259**	30,178
Narrow	2	**26,258**	**58,693**	58,681

TABLE V: Comparison of coverage objective and convergence time values observed for the DBS with fixed and variable steps.

Boosting Method	$(N = 8)$	$H(s^*)$		Convergence Time	
	Configuration	Fixed steps	Variable steps	Fixed steps	Variable steps
V-Boosting	General	140,592	140,649	550.7	91.3
	Room	127,557	127,517	613.5	140.3
	Maze	120,832	121,231	302.2	134.1
	Narrow	163,478	155,528	415.7	161.8
Arc-Boosting	General	140,615	140,542	80.3	104.9
	Room	127,647	127,455	390.0	158.1
	Maze	119,967	121,231	151.7	125.0
	Narrow	155,641	155,485	127.3	88.1
Average:		137,041	136,205	328.9	125.4

We complete this section by briefly addressing the effects of decentralization which in our simulation results show the DBS outperforming the CBS in every aspect. When all simulations carried out for $N = 10$ were considered (given in table II), on average (per simulation), the convergence time to the final optimal solution was improved (i.e., reduced) by 39.97%

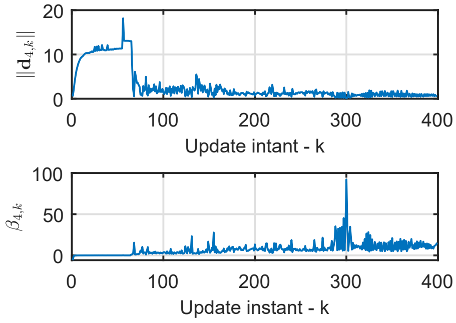


Fig. 10: Variation of gradient magnitude and the step size for agent $i = 4$ during the simulation which yielded Fig. 6h.

(approximately 165.2 s) due to the distributed nature of the DBS relative to a centralized approach. Further, due to decentralization, on average (per simulation), the final coverage cost achieved was increased by 0.381% (approximately 451 units). Finally, decentralization has the inherent advantages of reducing communication and implementation costs compared to a centralized solution.

V. Conclusions and Future Work

The concept of boosting functions provides a systematic approach to overcome the problem of multiple local optima arising in cooperative multi-agent optimization problems with non-convex objective functions. An optimal step size selection scheme was developed to guarantee the convergence of a distributed boosting scheme (DBS) for such general multi-agent optimization problems. The use of boosting functions in this DBS was demonstrated using the class of cooperative multi-agent coverage control problems, including two novel boosting function families. Simulation results are used to illustrate the effectiveness of the proposed boosting function families and the DBS. Ongoing research aims to explore the applicability of boosting functions in overcoming local optima in dynamic multi-agent optimization problems. Also, concerning coverage control applications, current research aims to explore how the constructed boosting function families should be modified if the objective function includes a cost term penalizing an agent team size.

References

[1] M. Zhong and C. G. Cassandras, "Distributed Coverage Control and Data Collection with Mobile Sensor Networks," *IEEE Trans. on Automatic Control*, vol. 56, no. 10, pp. 2445–2455, 2011.

[2] Z. Lin, L. Wang, Z. Han, and M. Fu, "Distributed Formation Control of Multi-Agent Systems Using Complex Laplacian," *IEEE Trans. on Automatic Control*, vol. 59, no. 7, pp. 1765–1777, 2014.

[3] J. Xu, C. Tekin, S. Zhang, and M. van der Schaar, "Distributed Multi-Agent Online Learning Based on Global Feedback," *IEEE Trans. on Signal Processing*, vol. 63, no. 9, pp. 2225–2238, 2015.

[4] C. Sun, S. Welikala, and C. G. Cassandras, "Optimal Composition of Heterogeneous Multi-Agent Teams for Coverage Problems with Performance Bound Guarantees," *Automatica*, vol. 117, p. 108961, 2020.

[5] N. Zhou, X. Yu, S. B. Andersson, and C. G. Cassandras, "Optimal Event-Driven Multi-Agent Persistent Monitoring of a Finite Set of Data Sources," *IEEE Trans. on Automatic Control*, vol. 63, no. 12, pp. 4204–4217, 2018.

[6] S. Su and Z. Lin, "Distributed Consensus Control of Multi-agent Systems with Higher Order Agent Dynamics and Dynamically Changing Directed Interaction Topologies," *IEEE Trans. on Automatic Control*, vol. 61, no. 2, pp. 515–519, 2015.

[7] M. Dotoli, S. Hammadi, K. Jeribi, C. Russo, and H. Zgaya, "A Multi-Agent Decision Support System for Optimization of Co-Modal Transportation Route Planning Services," in *Proc. of 52nd IEEE Conf. on Decision and Control*, 2013, pp. 911–916.

[8] D. K. Molzahn, F. Dorfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A Survey of Distributed Optimization and Control Algorithms for Electric Power Systems," *IEEE Trans. on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[9] A. Nedić and J. Liu, "Distributed Optimization for Control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 77–103, 2018.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[11] N. Bastianello, R. Carli, L. Schenato, and M. Todescato, "A Partition-Based Implementation of the Relaxed ADMM for Distributed Convex Optimization over Lossy Networks," in *Proc. of 57th IEEE Conf. on Decision and Control*, 2018, pp. 3379–3384.

[12] X. Sun, C. G. Cassandras, and K. Gokbayrak, "Escaping Local Optima in A Class of Multi-Agent Distributed Optimization Problems: A Boosting Function Approach," in *Proc. of 53rd IEEE Conf. on Decision and Control*, 2014, pp. 3701–3706.

[13] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[14] P. Chiu and F. Lin, "A Simulated Annealing Algorithm to Support the Sensor Placement for Target Location," in *Proc. of IEEE Canadian Conf. on Electrical and Computer Engineering*, 2004, pp. 867–870.

[15] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proc. of IEEE Intl. Conf. on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[16] J. H. Holland, "Genetic Algorithms and Adaptation," in *Adaptive Control of Ill-Defined Systems*. Springer US, 1984, pp. 317–333.

[17] M. Schwager, F. Bullo, D. Skelly, and D. Rus, "A Ladybug Exploration Strategy for Distributed Adaptive Coverage Control," in *Proc. of IEEE Intl. Conf. on Robotics and Automation*, 2008, pp. 2346–2353.

[18] P. Bianchi and J. Jakubowicz, "Convergence of A Multi-Agent Projected Stochastic Gradient Algorithm for Non-Convex Optimization," *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 391–405, 2013.

[19] T. Tatarenko and B. Touri, "Non-Convex Distributed Optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3744–3757, 2017.

[20] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, "How to Escape Saddle Points Efficiently," *Proc. of 34th Intl. Conf. on Machine Learning*, vol. 4, pp. 2727–2752, 2017.

[21] S. S. Du, C. Jin, J. D. Lee, M. I. Jordan, B. Póczos, and A. Singh, "Gradient Descent Can Take Exponential Time to Escape Saddle Points," in *Proc. of Advances in Neural Information Processing Systems*, 2017, pp. 1068–1078.

[22] X. Sun, C. G. Cassandras, and X. Meng, "A Submodularity-Based Approach for Multi-Agent Optimal Coverage Problems," in *Proc. of 56th IEEE Conf. on Decision and Control*, 2017, pp. 4082–4087.

[23] C. Sun, S. Welikala, and C. G. Cassandras, "Optimal Composition of Heterogeneous Multi-Agent Teams for Coverage Problems with Performance Bound Guarantees," *Automatica*, 2019.

[24] B. Addis, M. Locatelli, and F. Schoen, "Local Optima Smoothing for Global Optimization," *Optimization Methods and Software*, vol. 20, no. 4-5, pp. 417–437, 2005.

[25] J. D. Lee, I. Panageas, G. Piliouras, M. Simchowitz, M. I. Jordan, and B. Recht, "First-order Methods Almost Always Avoid Saddle Points," *Mathematical Programming*, vol. 176, no. 1-2, pp. 311–337, 2017.

[26] I. Panageas, G. Piliouras, and X. Wang, "First-Order Methods Almost Always Avoid Saddle Points: The Case of Vanishing Step-Sizes," 2019. [Online]. Available: http://arxiv.org/abs/1906.07772

[27] S. Welikala and C. G. Cassandras, "Distributed Non-convex Optimization of Multi-agent Systems Using Boosting Functions to Escape Local Optima," in *Proc. of American Control Conf.*, 2020.

[28] ——, "Distributed Non-Convex Optimization of Multi-Agent Systems Using Boosting Functions to Escape Local Optima: Theory and Applications," 2019. [Online]. Available: http://arxiv.org/abs/1903.04133

[29] C. A. Floudas and C. E. Gounaris, "A Review of Recent Advances in Global Optimization," *Journal of Global Optimization*, vol. 45, no. 1, pp. 3–38, 2008.

[30] J. S. Arora, O. A. Elwakeil, A. I. Chahande, and C. C. Hsieh, "Global Optimization Methods for Engineering Applications: A Review," *Structural Optimization*, vol. 9, no. 3-4, pp. 137–159, 1995.

[31] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2016.

[32] H. Flanders, "Differentiation Under the Integral Sign," *The American Mathematical Monthly*, vol. 80, no. 6, pp. 615–627, 1973.

Shirantha Welikala (S'13) received the B.Sc. Eng. degree in electrical and electronic engineering from the University of Peradeniya, Sri Lanka, in 2015. From 2015 to 2017, he was with the Department of Electrical and Electronic Engineering, University of Peradeniya, Sri Lanka, where he worked first as a Temporary Instructor and subsequently as a Research Assistant. He is currently pursuing the Ph.D. degree in Systems Engineering with Boston University, Brookline, MA, USA. His main research interests include optimization and control of multi-agent systems, robotics, signal processing, and smart grid applications.

Christos G. Cassandras (F'96) received the B.S. degree in engineering and applied science from Yale University, New Haven, CT, USA, in 1977, the M.S.E.E. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 1978, and the M.S. and Ph.D. degrees in applied mathematics from Harvard University, Cambridge, MA, USA, in 1979 and 1982, respectively. He was with ITP Boston, Inc., Cambridge, from 1982 to 1984, where he was involved in the design of automated manufacturing systems. From 1984 to 1996, he was a faculty member with the Department of Electrical and Computer Engineering, University of Massachusetts Amherst, Amherst, MA, USA. He is currently a Distinguished Professor of Engineering with Boston University, Brookline, MA, USA, the Head of the Division of Systems Engineering, and a Professor of Electrical and Computer Engineering. He serves on several editorial boards and has been a Guest Editor for various journals. 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 authored over 400 refereed papers in these areas, and six books. Dr. Cassandras is a Member of Phi Beta Kappa and Tau Beta Pi. He is also a Fellow of the International Federation of Automatic Control (IFAC). He was a recipient of several awards, including the 2011 IEEE Control Systems Technology Award, the 2006 Distinguished Member Award of the IEEE Control Systems Society, the 1999 Harold Chestnut Prize (IFAC Best Control Engineering Textbook), 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 was the Editor-in-Chief of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL from 1998 to 2009. He was the President of the IEEE Control Systems Society in 2012.