

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 in cooperative multi-agent optimization problems with non-convex objective functions. We propose a systematic approach to escape these local optima using boosting functions. These functions temporarily transform a gradient at a local optimum into a “boosted” non-zero gradient. Extending a prior centralized optimization approach, we develop a distributed framework for the use of boosted gradients and show that convergence of this distributed process can be attained by employing an optimal variable step size scheme for gradient-based algorithms. Numerical examples are included to show how the performance of a class of multi-agent optimization systems can be improved.

I. INTRODUCTION

A cooperative multi-agent system is a collection of interacting subsystems (also called *agents*), where each agent controls its *local state* so as to collectively optimize a common *global objective* subject to various constraints. In a *distributed* optimization approach, each agent controls its state using only locally available information. The goal is to drive all agents to a globally optimal set of states. This can be a challenging task depending on the nature of: (i) the agents (which may be sensor nodes, vehicles, robots, supply sources, or processors of a multi-core computer), (ii) the constraints on their decision space, (iii) the inter-agent interactions, and, (iv) the global objective function. Therefore, a large number of optimization methods can be found in the literature specifically developed to address different classes of multi-agent systems.

Cooperative multi-agent system optimization arises in coverage control [1], formation control [2], monitoring [3], flocking [4], resource allocation [5], learning [6], consensus [7], transportation [8] and smart grid [9]. In these applications, gradient-based techniques are typically used due to their simplicity (see the survey paper [10]). However, more computationally complex schemes, e.g., using the Alternating Direction Method of Multipliers (ADMM) [11], are also gaining popularity due to their greater generality.

In some multi-agent systems, properties of the associated objective function, such as convexity, can be exploited to achieve a global optimum. For example, the Relaxed-ADMM approach in [11] converges to the global optimum for convex objective functions. On the other hand, there are many settings where the objective function takes a non-convex form making it difficult to attain a global optimum

[1], [12]. In such situations, one often resorts to global optimization techniques such as simulated annealing [13], genetic algorithms [14], or particle swarm algorithms [15]. The common feature in these approaches is to introduce an element of randomness in the process of controlling agents. These methods are computationally intensive and usually infeasible for on-line optimization.

The issue of non-convexity in the objective functions has recently attracted renewed attention for specific classes of multi-agent systems by exploiting properties that the objective function may possess. For example, when the objective function is submodular, tight performance bound guarantees may be found [16]. Methods like local optima smoothing [17] and balanced detection [1] trade-off local approximations and global exploration of the objective function to achieve a better optimum. In [12], the concept of a “boosting function” is used to escape local optima and seek better ones through an exploration of the search space which exploits the objective function’s structure. However, none of these methods so far is designed to function in a *distributed* multi-agent setting and convergence guarantees are lacking.

In this paper, we propose a distributed approach to solve general non-convex multi-agent optimization problems, based on the centralized boosting function approach in [12]. The key idea is to temporarily alter the local objective function of an agent whenever an equilibrium is reached, by defining a new auxiliary local objective function. This is done indirectly by transforming the local gradient (of the local objective) to get a new *boosted gradient* (which corresponds to the gradient of the unknown auxiliary local objective). Therefore, a boosting function, formally, is a transformation of the local gradient, whenever it becomes zero; the result of the transformation is a non-zero *boosted gradient*. After following the boosted gradient, when a new equilibrium point is reached, we revert to the original objective function and the gradient-based algorithm converges to a new (potentially better and never worse) equilibrium point. In contrast to randomly perturbing the gradient components (e.g., [13]), boosting functions provide a systematic way to force each agent to move in a well-chosen direction that further explores the feasible space based on structural properties of the objective function and on knowledge of both the feasible space and of the current agent states. Details on the design of boosting functions and their use in the proposing distributed optimization framework of this paper are given in [18].

The contribution of this paper is to first provide a formal analysis of the original centralized boosting scheme [12] so as to establish convergence and then develop a distributed scheme whereby each agent may asynchronously switch

*Supported in part by NSF under grants ECCS-1509084, DMS-1664644, CNS-1645681, by AFOSR under grant FA9550-19-1-0158, by ARPA-E’s NEXTCAR program under grant DE-AR0000796 and by the MathWorks.

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

between a boosting and a normal mode independent of other agents. We show that the latter scheme also converges, i.e., the asynchronous boosting processes reach a terminal point where a new (generally local but improved) optimum is reached. Central to this process is a method for selecting *optimal variable step sizes* in the underlying distributed gradient-based optimization algorithm.

II. PROBLEM FORMULATION

We consider cooperative multi-agent optimization problems of the general form,

$$\mathbf{s}^* = \arg \max_{\mathbf{s} \in \mathbf{F}} H(\mathbf{s}), \quad (1)$$

where, $H: \mathbb{R}^{mN} \rightarrow \mathbb{R}$ is the *global objective function* and $\mathbf{s} = [s_1, s_2, \dots, s_N] \in \mathbb{R}^{mN}$ is the *controllable global state*. Here, for any $i \in \{1, 2, \dots, N\}$, $s_i \in \mathbb{R}^m$ represents the *local state* of agent i . Further, $\mathbf{F} \subseteq \mathbb{R}^{mN}$ represents the feasible space for $\mathbf{s}$. In this work, linearity or convexity-related conditions are not imposed on the global objective function $H(\mathbf{s})$.

In order to model the inter-agent interactions, an undirected graph denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ is used where $\mathcal{V} = \{1, 2, \dots, N\}$ is a set of N agents, and, $\mathcal{A}$ is the set of communication links between those agents. The set of *neighbors* of an agent $i \in \mathcal{V}$ is denoted by $B_i = \{j : j \in \mathcal{V}, (i, j) \in \mathcal{A}\}$. The *closed neighborhood* of agent i is defined as $\bar{B}_i = B_i \cup \{i\}$ and $|\bar{B}_i|$ denotes the cardinality of the set $\bar{B}_i$. 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, an agent i is also assumed to have a *local objective function* $H_i(\bar{s}_i)$ where $H_i: \mathbb{R}^{m|\bar{B}_i|} \rightarrow \mathbb{R}$. 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. For example, two common possibilities are the *additive form* [11] $H(\mathbf{s}) = \sum_{i=1}^N H_i(\bar{s}_i)$ and the *separable form* [12] $H(\mathbf{s}) = H_i(\bar{s}_i) + H_i^c(s_i^c)$ with $H_i^c: \mathbb{R}^{m(N-1)} \rightarrow \mathbb{R}$ and $s_i^c = [s_1, s_2, \dots, s_{i-1}, s_{i+1}, \dots, s_N]$. The latter includes a large class of common multi-agent problems studied in [1].

Due to the versatile nature of H and $\mathbf{F}$ in (1), applicable solving techniques are limited to global optimization methods. Even-though many such techniques are available [15], in this paper we consider a simple gradient-ascent scheme so as to take advantage of its simplicity in terms of analysis, computation, and on-line implementation, despite the obvious limitation of attaining only local optima. We are also interested in solving (1) through *distributed* schemes so that each agent i updates its local state s_i according to

$$s_{i,k+1} = s_{i,k} + \beta_{i,k} d_{i,k}, \quad (2)$$

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

A. Escaping local optima using boosting functions

Converging to a local optimum is the main drawback of using a gradient-based method like (2), when the global objective function H is non-convex and/or the feasible space $\mathbf{F}$ is non-convex. In [12], this problem has been addressed by

introducing the concept of *boosting functions* as an effective systematic method of escaping local optima.

Boosting functions. The main idea here is to temporarily alter the local objective function $H_i(\bar{s}_i)$ whenever an equilibrium is reached with a newly defined auxiliary objective function $\hat{H}_i(\bar{s}_i)$. However, we are interested in the *boosted gradient* $\hat{d}_{i,k} = \frac{\partial \hat{H}_i(\bar{s}_i)}{\partial s_i}$ rather than $\hat{H}_i(\bar{s}_i)$. A boosted gradient is a transformation of the associated local gradient d_i taking place at an equilibrium point (where its value is zero); the result of the transformation is a non-zero $\hat{d}_i \neq 0$ which, therefore, forces the agent to move in a direction determined by the boosting function and to further explore the feasible space. When a new equilibrium point is reached, we revert to the original objective function and then 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 the objective functions $H(\mathbf{s})$ and $H_i(\bar{s}_i)$, of the feasible space $\mathbf{F}$, and of the agent state trajectories. Unlike various forms of randomized state perturbations away from their current equilibrium [13], boosting functions provide a formal rational systematic transformation process of the form $\hat{d}_i = f(d_i, \bar{s}_i)$ where the boosting function f depends on the specific problem type. More details on boosting functions including a generic set of guidelines on selecting them are given in [18]. As an example, in coverage control problems local optima arise when a cluster of agents provides high-quality local coverage in a region while ignoring other regions; in this case, a boosting function that enhances separation between close neighbors is an intuitive choice that has been shown to be effective [12].

Boosting scheme. When an agent i is following the boosted gradient direction $\hat{d}_{i,k}$, it is said to be in the *Boosting Mode* and its state updates take the form

$$s_{i,k+1} = s_{i,k} + \beta_{i,k} \hat{d}_{i,k}. \quad (3)$$

Similarly, when an agent i is following the "normal" gradient direction $d_{i,k}$ as in (2), 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 normal mode is denoted by $\mathbf{N}$ and the boosting mode is denoted by $\mathbf{B}$. 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 $\mathbf{B}$ is entered when $d_i = 0$ for all $i \in \mathcal{V}$. After $\hat{d}_i = 0$ for all $i \in \mathcal{V}$, the agents re-enter mode $\mathbf{N}$ 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 $\mathbf{B}$ 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 $\mathbf{B}$ to obtain improved performance for a variety of multi-agent coverage control prob-

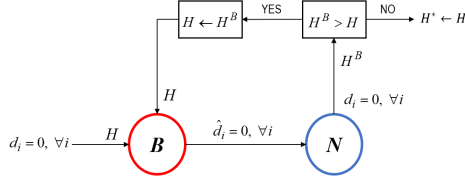


Fig. 1: A centralized boosting scheme (CBS)

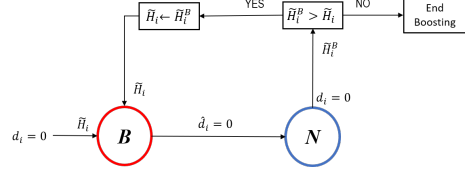


Fig. 2: A distributed boosting scheme (DBS) asynchronously applied by each agent $i = 1, \dots, N$

lems. However, there has been no formal proof to date that this process actually converges. Moreover, our goal is to develop a *Distributed Boosting Scheme* (DBS) where each agent can independently switch between modes **B** and **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, in problems such as coverage control [1], the original problem is inherently distributed and makes a DBS a natural approach.

A 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 $\tilde{H}_i$, which will be formally introduced later (in (10)). Additional implementation details of the DBS (including their algorithmic constructions) are discussed in [18]. However, they do not affect the convergence analysis that follows. 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 (2) and (3).

Convergence criteria. When a DBS is considered, unlike the case of a CBS in Fig. 1, the decentralized nature of agent behavior causes agents to switch between modes (normal/boosting) independently and asynchronously from each other. As a result, at a given time instant, a subset of the agents will be in normal mode (following (2)) while others are in boosting mode (following (3)). This partition of the complete agent set $\mathcal{V}$ leads to two agent sets henceforth denoted by $\mathcal{N}$ and $\mathcal{B}$ respectively. For any agent $i \in \mathcal{V}$, the following conditions are defined as the *convergence criteria*:

$$\lim_{k \rightarrow \infty} d_{i,k} = 0 \text{ when } \bar{B}_i \subseteq \mathcal{N}, \quad (4)$$

$$\lim_{k \rightarrow \infty} d_{i,k} = 0 \text{ when } i \in \mathcal{N}, B_i \cap \mathcal{B} \neq \emptyset, \quad (5)$$

$$\lim_{k \rightarrow \infty} \hat{d}_{i,k} = 0 \text{ when } i \in \mathcal{B}, B_i \cap \mathcal{B} \neq \emptyset. \quad (6)$$

Since each agent i , irrespective of its neighbors' modes, continuously switches between the normal and boosting modes, boosting will only continue as long as there is a

gain from the boosting stages (i.e., $\tilde{H}_i^B > \tilde{H}_i$ in Fig. 2). It is then clear how these criteria can guarantee convergence.

Upon termination (i.e., all agents reached “End Boosting”) of the boosting scheme, achieving $d_{i,k} = 0, \forall i \in \mathcal{V}$ is guaranteed by the criterion (4). In this work, it is assumed that the relationship between $H(\mathbf{s})$ and $H_i(\bar{s}_i)$, $i \in \mathcal{V}$ is such that $d_{i,k} = 0, \forall i \in \mathcal{V} \implies \nabla H(\mathbf{s}_k) = 0$. Clearly this assumption holds for any problem with a separable form where $H(\mathbf{s}) = H_i(\bar{s}_i) + H_i^c(s_i^c)$ and for most problems of interest with an additive form where $H(\mathbf{s}) = \sum_{i=1}^N H_i(\bar{s}_i)$ (more details can be found in [18]). Therefore, achieving (4)-(6) directly implies that $\nabla H(\mathbf{s}_k) = 0$ and convergence to a solution of (1) (again, not necessarily a global optimum) is obtained.

III. CONVERGENCE ANALYSIS THROUGH OPTIMAL VARIABLE STEP SIZES

As a means of enforcing convergence for a general problem of the form (1), a variable step size scheme is next proposed and shown to guarantee (4)-(6). Due to space limitations, this section focuses only on addressing the unconstrained version of (1) and all proofs are omitted but can be found in [18]. Our main results depend on a few assumptions, starting with the following conditions on the nature of the local objective functions.

Assumption 1: Any local objective function $H_i(\bar{s}_i)$, $i \in \mathcal{V}$, 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., $H_i(x) < H_{UB} < \infty$, $x \in \mathbb{R}^{m|\bar{B}_i|}$.

We begin by developing an optimal variable step size scheme for agents $i \in \mathcal{V}$ such that $\bar{B}_i \subseteq \mathcal{N}$ (i.e., all neighboring agents are also in normal mode - following (2)). The respective convergence criterion for this case is (4).

A. Convergence of agent i when $\bar{B}_i \subseteq \mathcal{N}$

For notational convenience, let $\mathbf{q}_i = \{1, 2, \dots, q_i\}$ with $q_i = |\bar{B}_i|$ represent 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 (2) for all $j \in \bar{B}_i$. Here, $\bar{s}_{i,k+1}$, $\bar{s}_{i,k}$ and $\bar{\mathbf{d}}_{i,k}$ are $m q_i$ -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 .

For a function $f: \mathbb{R}^n \rightarrow \mathbb{R}$, if the Lipschitz continuity constant of ∇f is L , the descent lemma [19] applies, and, its ascent version (Lemma 1 in [18]) is: $\forall x, y \in \mathbb{R}^n$, $f(x+y) \geq f(x) + y^T \nabla f(x) - \frac{L}{2} \|y\|^2$. Thus, under Assumption 1, the ascent lemma can be applied to a local objective function $H_i(\bar{s}_{i,k})$ for the state update $\bar{s}_{i,k+1} = \bar{s}_{i,k} + \bar{\beta}_{i,k} \bar{\mathbf{d}}_{i,k}$ as:

$$H_i(\bar{s}_{i,k+1}) \geq H_i(\bar{s}_{i,k}) + (\bar{\beta}_{i,k} \bar{\mathbf{d}}_{i,k})^T \nabla H_i(\bar{s}_{i,k}) - \frac{K_{1i}}{2} \|\bar{\beta}_{i,k} \bar{\mathbf{d}}_{i,k}\|^2.$$

Therefore,

$$H_i(\bar{s}_{i,k+1}) \geq H_i(\bar{s}_{i,k}) + \sum_{j \in \bar{B}_i} \Delta_{ji,k}, \quad (7)$$

$$\Delta_{ji,k} \triangleq \beta_{j,k} d_{j,k}^T d_{ji,k} - \frac{K_{1i}}{2} \beta_{j,k}^2 \|d_{j,k}\|^2 \in \mathbb{R}, \quad (8)$$

$$d_{ji,k} \triangleq \nabla_j H_i(\bar{s}_{i,k}) = \frac{\partial H_i(\bar{s}_{i,k})}{\partial s_j} \in \mathbb{R}^m. \quad (9)$$

The term $d_{ji,k}$ in (9) 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 (8) depends on the step size $\beta_{j,k}$ which is selected by agent $j \in \bar{B}_i$. In (7), each $\Delta_{ji,k}$ term can be thought of as a contribution coming from neighboring agent j to agent i , so as to improve (increase) H_i . However, in order for an 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 $i \in \mathcal{V}$ has knowledge of the cross-gradient terms $\{d_{ij,k}, j \in \bar{B}_i\}$, neighbor objective function values $\{H_j, j \in \bar{B}_i\}$ and the local Lipschitz constants $\{K_{1j}, j \in \bar{B}_i\}$ at the k^{th} update instant.

This assumption is consistent with our concept of neighborhood, where neighbors share information through communication links. Thus, any agent i has access to the parameters it requires: $d_{ij,k} (= \partial H_j / \partial s_i)$, H_j and K_{1j} from all its neighbors $j \in \bar{B}_i$. 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 exchanges. Many cooperative multi-agent optimization problems have this structure including the class of multi-agent coverage control problems as shown in [18].

We now define a *neighborhood objective* function $\tilde{H}_i$ for any $i \in \mathcal{V}$, where $\tilde{H}_i : \mathbb{R}^{m|\bar{B}_i|} \rightarrow \mathbb{R}$ and $\bar{B}_i = \cup_{j \in \bar{B}_i} B_j$, as

$$\tilde{H}_{i,k} = \sum_{j \in \bar{B}_i} H_j(\bar{s}_{j,k}). \quad (10)$$

This neighborhood objective function value can be viewed as agent i 's estimate of the total contribution of agents in $\bar{B}_i$ towards the global objective function. Note that it can be computed locally (i.e., at i) under Assumption 2.

Remark 1: The neighborhood objective functions play an important role in a DBS because a distributed scheme comes at the cost of each agent losing the global information $H(\mathbf{s})$. In contrast, in the CBS of Fig. 1 $H(\mathbf{s})$ plays a crucial role in the " $H^B > H$ " block. As a remedy, 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. However, as seen in the ensuing analysis, the form of $\tilde{H}_i$ is not limited to (10) - it can take any appropriate form (see [18] for details).

By writing (7) for an agent j 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 $j \in B_i$ and using the definition in (10) yields

$$\tilde{H}_{i,k+1} \geq \tilde{H}_{i,k} + (\tilde{\Delta}_{i,k} + Q_{i,k}), \quad (11)$$

$$\tilde{\Delta}_{i,k} \triangleq \sum_{j \in \bar{B}_i} \Delta_{ij,k}, \quad (12)$$

$$Q_{i,k} \triangleq \sum_{j \in B_i} (\Delta_{jj,k} + \Delta_{ji,k}) + \sum_{l \in \bar{B}_i - \{i\}} \Delta_{lj,k}. \quad (13)$$

Note that $\tilde{\Delta}_{i,k}$ in (12) is a function of terms $\Delta_{ij,k}$ (and not $\Delta_{ji,k}$) which are locally available to and controlled by agent i , i.e., $\beta_{i,k}$, $d_{i,k}$ and $d_{ij,k}, \forall j \in \bar{B}_i$. In contrast, agent i does not have any control over $Q_{i,k}$ in (13), as this strictly depends through (8) on the step sizes of agent i 's *extended neighborhood*, i.e., $\beta_{j,k}, \forall j \in \bar{B}_i - \{i\}$.

Nonetheless, (11) implies that the neighborhood objective function $\tilde{H}_{i,k}$ can be increased by at least $(\tilde{\Delta}_{i,k} + Q_{i,k})$ at any update instant k . Thus, to maximize the gain in $\tilde{H}_{i,k}$, agent i 's step size $\beta_{i,k}$ is selected according to the *auxiliary problem*:

$$\beta_{i,k}^* = \arg \max_{\beta_{i,k}} \tilde{\Delta}_{i,k} \quad (14)$$

subject to $\tilde{\Delta}_{i,k} > 0$.

Lemma 1: The solution to the auxiliary problem (14) is

$$\beta_{i,k}^* = \frac{1}{\sum_{j \in \bar{B}_i} K_{1j}} \frac{d_{i,k}^T \sum_{j \in \bar{B}_i} d_{ij,k}}{\|d_{i,k}\|^2}. \quad (15)$$

Let us denote the optimal objective function value of (14) as $\tilde{\Delta}_{i,k}^*$. It is easy to show that $\beta_{i,k}^*$ in (15) is feasible (i.e., $\tilde{\Delta}_{i,k}^* > 0$) as long as $\beta_{i,k}^* \neq 0$. 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 , if it occurs, the agent can consider to 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 $\tilde{\Delta}_{i,k}^* > 0$.

To establish the convergence proof in Theorem 1, we need the following two assumptions.

Assumption 3: Consider the sum,

$$\tilde{Q}_{i,k} = \sum_{l=k-T_i}^k Q_{i,l}, \quad (16)$$

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

When the graph $\mathcal{G}(\mathcal{V}, \mathcal{A})$ is complete, Assumption 3 is immediately satisfied with $T_i = 1, \forall i \in \mathcal{V}$ (see also [18]).

Assumption 4: For all $i \in \mathcal{V}$, there exists a function $\Psi_{i,k}$ such that $0 < \Psi_{i,k}$ and

$$\begin{cases} 0 \leq \Psi_{i,k} \|d_{i,k}\|^2 \leq \tilde{\Delta}_{i,k}^* + \tilde{Q}_{i,k}, & \text{when } 0 < \tilde{\Delta}_{i,k}^* + \tilde{Q}_{i,k}, \\ 0 \leq \Psi_{i,k} \|d_{i,k}\|^2 \leq \tilde{\Delta}_{i,k}^*, & \text{when } 0 < \tilde{\Delta}_{i,k}^*. \end{cases} \quad (17)$$

$$\quad (18)$$

This assumption is trivial because whenever the optimal step size in (15) 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 always a candidate function for $\Psi_{i,k}$. And when $0 < \tilde{\Delta}_{i,k}^* + \tilde{Q}_{i,k}$ occurs, $\Psi_{i,k} = (\tilde{Q}_{i,k} + \tilde{\Delta}_{i,k}^*) / (K_2 \|d_{i,k}\|^2)$ can be used as a candidate function.

Theorem 1: For all $i \in \mathcal{V}$ such that $\bar{B}_i \subseteq \mathcal{N}$, under Assumptions 1, 2, 3, and 4, the step size selection in (15) guarantees the convergence criterion (4), i.e., $\lim_{k \rightarrow \infty} d_{i,k} = 0$.

B. Convergence of agent i when $B_i \cap \mathcal{B} \neq \emptyset$

In this case, at least some of the agents in $\bar{B}_i$ are in boosting mode, following (3). Following the same approach as in Section III-A, we seek an optimal variable step size selection scheme similar to (15) so as to ensure the convergence criteria given in (5) and (6). Compared to (7) the ascent lemma relationship for $H_i(\bar{s}_{i,k})$ takes the form:

$$H_i(\bar{s}_{i,k+1}) \geq H_i(\bar{s}_{i,k}) + \sum_{j \in \bar{B}_i \cap \mathcal{N}} \Delta_{ji,k} + \sum_{j \in \bar{B}_i \cap \mathcal{B}} \hat{\Delta}_{ji,k}, \quad (19)$$

where $\Delta_{ji,k}$ for $j \in \mathcal{N}$ is the same as (8) and we set

$$\hat{\Delta}_{ji,k} = \beta_{j,k} \hat{d}_{j,k}^T d_{ji,k} - \frac{K_{1i}}{2} \beta_{j,k}^2 \|\hat{d}_{j,k}\|^2 \in \mathbb{R}. \quad (20)$$

Then, the ascent lemma for neighborhood objective function $\tilde{H}_{i,k}$ can be expressed as: $\tilde{H}_{i,k+1} \geq \tilde{H}_{i,k} + (\tilde{\Delta}_{i,k} + Q_{i,k})$ with

$$\tilde{\Delta}_{i,k} \triangleq 1_{\{i \in \mathcal{N}\}} \left[\sum_{j \in \mathcal{B}_i} \Delta_{ij(k)} \right] + 1_{\{i \in \mathcal{B}\}} \left[\sum_{j \in \mathcal{B}_i} \hat{\Delta}_{ij(k)} \right], \quad (21)$$

$$Q_{i,k} \triangleq \sum_{j \in \mathcal{B}_i} (1_{\{j \in \mathcal{N}\}} [\Delta_{jj,k} + \Delta_{ji,k}] + 1_{\{j \in \mathcal{B}\}} [\hat{\Delta}_{jj,k} + \hat{\Delta}_{ji,k}]) + \sum_{l \in \{B_j - \{i\}\}} [1_{\{l \in \mathcal{N}\}} \Delta_{lj,k} + 1_{\{l \in \mathcal{B}\}} \hat{\Delta}_{lj(k)}], \quad (22)$$

where $1_{\{\cdot\}}$ is the usual indicator function. Under this new $\tilde{\Delta}_{i,k}$ in (21), the same auxiliary problem as in (14) is used to determine the step size $\beta_{i,k}^*$ to optimally increase the neighborhood cost function $\tilde{H}_{i,k}$.

Lemma 2: The solution to the auxiliary problem (14) with $\tilde{\Delta}_{i,k}$ given in (21) is

$$\beta_{i,k}^* = \begin{cases} \frac{1}{\sum_{j \in \mathcal{B}_i} K_{1j}} \frac{d_{i,k}^T (\sum_{j \in \mathcal{B}_i} d_{ij,k})}{\|d_{i,k}\|^2} & \text{when } i \in \mathcal{N}, \\ \frac{1}{\sum_{j \in \mathcal{B}_i} K_{1j}} \frac{\hat{d}_{i,k}^T (\sum_{j \in \mathcal{B}_i} \hat{d}_{ij,k})}{\|\hat{d}_{i,k}\|^2} & \text{when } i \in \mathcal{B}. \end{cases} \quad (23)$$

Note that the result in (23) is a generalization of (15).

To establish the convergence criteria (5) and (6), Assumptions 1, 2, 3 and 4 are still required. However, note that Assumption 3 should now be considered under the new expression for $Q_{i,k}$ in (22) and $d_{i,k}$ terms used in Assumption 4 should now be replaced by $[1_{\{i \in \mathcal{N}\}} d_{i,k} + 1_{\{i \in \mathcal{B}\}} \hat{d}_{i,k}]$.

Theorem 2: Under Assumptions 1,2,3, and 4, the step size selection in (23) guarantees the convergence conditions stated in (4)-(6): if $i \in \mathcal{N}$, then $\lim_{k \rightarrow \infty} d_{i,k} = 0$, and, if $i \in \mathcal{B}$, then $\lim_{k \rightarrow \infty} \hat{d}_{i,k} = 0$.

Remark 2: The following issues are addressed and extensively discussed in [18]: (i) Extending the proposed technique to handle time varying neighborhoods, (ii) Use of projections to handle the feasible space (F) constraint (and its convergence), and, (iii) Advantages of using the proposed variable step sizes (23) compared to using fixed step sizes.

IV. AN APPLICATION EXAMPLE

In this section, we will apply the DBS shown in Fig. 2 combining (2) and (3) with step sizes given through (23) to a class of multi-agent systems with an objective function $H(s)$ in (1) of the form

$$H(s) = \int_F R(x) P(x, s) dx. \quad (24)$$

where $F \subset \mathbb{R}^2$ is a (generally non-convex) feasible region. The function $R: F \rightarrow \mathbb{R}$ provides a “value” assigned to each point $x \in \mathbb{R}^2$ with the properties: $R(x) = 0$ if $x \notin F$, $R(x) \geq 0$ if $x \in F$ and $\int_F R(x) dx < \infty$. The function $P(x, s)$ captures the reward incurred when the system under state $s = [s_1, s_2, \dots, s_N] \in \mathbb{R}^{mN}$ interacts with $x \in \mathbb{R}^2$ (e.g., the reward from collecting data from x if this corresponds to a data source.) It is easy to see that many problems of interest involving multi-agent systems can be expressed in the form (24), including coverage control and a variety of consensus-based problems [1]. Here, we limit ourselves to coverage

control problems, so that $R(x)$ is an *event density* function ($R(x) = 1, \forall x \in F$ if events occur uniformly over F) and $P(x, s)$ is the *joint probability* of detecting an event occurring at $x \in F$. Assuming independently detecting agents, $P(x, s) = 1 - \prod_{i=1}^N [1 - \hat{p}_i(x, s_i)]$ with $\hat{p}_i(x, s_i) = 1_{\{x \in V_i\}} p_i(x, s_i)$ where $V_i = \{x: \|x - s_i\| \leq \delta_i, \forall \lambda \in (0, 1), (\lambda x + (1 - \lambda)s_i) \in F\}$ is the *visibility region* of agent i limited by its sensing range δ_i and physical obstacles (see Fig. 3) and $p_i(x, s_i)$ represents the probability that agent i (with local state s_i) senses an event at $x \in F$. In this setting, we view $s_i \in F$ as simply the location of the agent in a two-dimensional “mission space.” It is assumed that $p_i(x, s_i)$ is a differentiable and monotonically decreasing function in $D_i = \|x - s_i\|$. Note that $\hat{p}_i(x, s_i)$ is discontinuous w.r.t. x or s_i because of the presence of obstacles.

In [12], for $H(s)$ in (24), it is shown that $H(s)$ has a decomposable form $H(s) = H_i(\bar{s}_i) + H_i^c(s_i^c)$ with $H_i(\bar{s}_i) = \int_{V_i} R(x) \prod_{j \in \mathcal{B}_i} [1 - \hat{p}_j(x, s_j)] p_i(x, s_i) dx$ and $H_i^c(s_i^c) = \int_F R(x) \{1 - \prod_{j \in \mathcal{V} - \{i\}} [1 - \hat{p}_j(x, s_j)]\} dx$. Note that $H_i(\bar{s}_i)$ depends only on the closed neighborhood of agent i and $H_i^c(s_i^c)$ is independent of s_i as s_i^c contains only states in the set $\{s_j: j \in \mathcal{V} - \{i\}\}$. The gradient d_i in (2) is a vector $d_i = [d_{iX}, d_{iY}]^T \in \mathbb{R}^2$. Using the Leibniz rule [20] and following some algebra (details are given in [18]), we can derive d_{iX} (and, similarly, d_{iY}) as

$$d_{iX} = \int_{V_i} w_{i1}(x, \bar{s}_i) \frac{(x - s_i)_X}{\|x - s_i\|} dx + \sum_{\Gamma_{ij} \in \Gamma_i} \text{sgn}(n_{ijX}) \frac{\sin \theta_{ij}}{\|v_{ij} - s_i\|} \int_0^{Z_{ij}} w_{i2}(\rho_{ir}(r), \bar{s}_i) r dr + \sum_{\Theta_{ij} \in \Theta_i} \delta_i \cos \theta \int_{\theta_{j1}}^{\theta_{j2}} w_{i3}(\rho_{i\theta}(\theta), \bar{s}_i) d\theta, \quad (25)$$

where $\text{sgn}(\cdot)$ is the signum function, $\rho_{ir}(r) = \frac{v_{ij} - s_i}{\|v_{ij} - s_i\|} r + v_{ij}$ and $\rho_{i\theta}(\theta) = s_i + \delta_i [\cos \theta \sin \theta]^T$. The last two terms of (25) arise due to the boundary ∂V_i of the sensing region which depends on δ_i (the sensing range) and obstacle vertices v_{ij} . Note that the presence of polygonal obstacles is assumed and all associated geometric parameters can be seen in Fig. 3. The segments of ∂V_i affecting (25) consist of two sets: $\Theta_i = \{\Theta_{i1}, \Theta_{i2}, \dots\}$ representing circular arc segments (due to δ_i), and $\Gamma_i = \{\Gamma_{i1}, \Gamma_{i2}, \dots\}$ representing linear segments (due to v_{ij} 's). Each Θ_{ij} term is a pair $(\theta_{j1}, \theta_{j2})$ with starting angle θ_{j1} and ending angle θ_{j2} . Similarly, each Γ_{ij} term is a 4-tuple $(Z_{ij}, \theta_{ij}, v_{ij}, n_{ij})$ with the geometric parameters: end point Z_{ij} , angle θ_{ij} , obstacle vertex v_{ij} , and unit normal direction n_{ij} as seen in Fig. 3. Finally, setting $\Phi_i(x) = \prod_{j \in \mathcal{B}_i} [1 - \hat{p}_j(x, s_j)]$, gives the weight functions as,

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

with $l = 1, 2$. Ignoring the details leading to (25), the crucial observation is that its three terms can be viewed as “forces” acting on agent i (located at s_i) and generated by points $x \in V_i$. The weight function $w_{i1}(x, \bar{s}_i)$ represents an attraction force towards point $x \in V_i$. The weight function $w_{i2}(x, \bar{s}_i)$ is the magnitude of a force generated in a direction lateral to the line Γ_{ij} (towards the interior of V_i) by a point $x \in \Gamma_{ij}$.

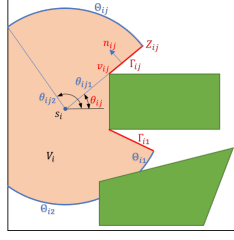


Fig. 3: Mission space with one agent.

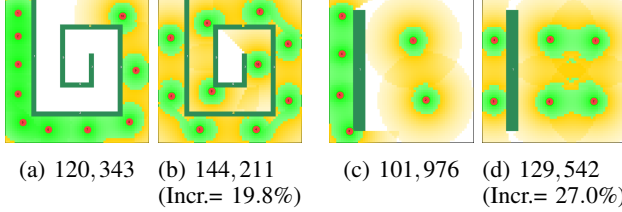


Fig. 4: Coverage improvement due to distributed boosting. ((a),(c)): Gradient ascent, ((b),(d)): Distributed Φ -boosting. Red dot locations show s^* . Captions give H^* and percentage coverage improvement due to boosting.

Similarly, $w_{i3}(x, \bar{s}_i)$ represents the magnitude of an attraction force generated by a point $x \in \Theta_{ij}$. Recalling that a boosting function is a transformation $\hat{d}_i = f(d_i, \bar{s}_i)$, we can now systematically define such transformations through

$$\hat{w}_{ij}(x, \bar{s}_i) = \alpha_{ij}(x, \bar{s}_i)w_{ij}(x, \bar{s}_i) + \eta_{ij}(x, \bar{s}_i), \quad j = 1, 2, 3 \quad (26)$$

where $\alpha_{ij}, \eta_{ij} : \mathbb{R}^2 \times \mathbb{R}^{2|\bar{B}_i|} \rightarrow \mathbb{R}$ are transformation functions we can select, typically nonlinear in their arguments. Thus, the boosted gradient constructed when $d_{i,k} = 0$ in (2) is some $\hat{d}_{i,k} \neq 0$ obtained from (25) by applying (26) to the original weights. Note that $d_{i,k} = 0$ occurs when all the aforementioned virtual forces add up to a resultant force with zero magnitude. Thus, the boosted weights $\hat{w}_{ij}(x, \bar{s}_i)$, $j = 1, 2, 3$, alter these forces in a way controlled by α_{ij}, η_{ij} and note that when $\alpha_{ij}(x, \bar{s}_i) = 1$ and $\eta_{ij}(x, \bar{s}_i) = 0$, the boosted gradient $\hat{d}_i$ reduces to the normal gradient d_i . Clearly, the linear form in (26) could be generalized.

Our purpose here is only to illustrate the use of the DBS in Fig. 2. Thus we limit ourselves to the Φ -Boosting function method introduced in [12] which uses,

$$\eta_{ij}(x, \bar{s}_i) = 0, \text{ and, } \alpha_{i1}(x, \bar{s}_i) = \kappa \Phi_i(x)^\gamma. \quad (27)$$

Note that $\Phi_i(x)$ indicates the extent to which point x is not covered by neighbors B_i . Thus, Φ -Boosting forces agent i to move towards regions of V_i which are less covered by its neighbors. A conventional gradient ascent method (which simply uses (2)) is compared with the distributed Φ -Boosting method (which uses both (2) and (3) according to the DBS in Fig. 2) in Fig. 4 in terms of s^* and H^* achieved for two different obstacle arrangements. Several more simulation results can be found in [18]. The reader is invited to reproduce these results and explore other boosting functions at <http://www.bu.edu/codes/simulations/shiran27/CoverageFinal/>.

V. CONCLUSION

The concept of boosting provides systematic ways 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 is developed to guarantee convergence in a distributed (or centralized) framework for such general multi-agent optimization problems. The new distributed boosting scheme is illustrated in simulation examples showing that boosting can considerably improve performance without significantly affecting the computational cost involved. Ongoing research aims to explore the generality of boosting functions to be used regardless of the intended application.

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] N. Zhou, X. Yu, S. B. Andersson, and C. G. Cassandras, "Optimal Event-Driven Multiagent Persistent Monitoring of a Finite Set of Data Sources," *IEEE Trans. on Automatic Control*, vol. 63, no. 12, pp. 4204–4217, Dec 2018.
- [4] S. Ghapani, J. Mei, W. Ren, and Y. Song, "Fully distributed flocking with a moving leader for Lagrange networks with parametric uncertainties," *Automatica*, vol. 67, pp. 67–76, 2016.
- [5] J. R. Marden and T. Roughgarden, "Generalized Efficiency Bounds in Distributed Resource Allocation," *IEEE Trans. on Automatic Control*, vol. 59, no. 3, pp. 571–584, 2014.
- [6] L. Panait and S. Luke, "Cooperative Multi-Agent Learning: The State of the Art," *Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 3, pp. 387–434, 2005.
- [7] H. Sun, Y. Liu, F. Li, and X. Niu, "A survey on optimal consensus of multi-agent systems," in *IEEE Chinese Automation Congress*, Oct 2017, pp. 4978–4983.
- [8] B. Burmeister, A. Haddadi, and G. Matylis, "Application of multi-agent systems in traffic and transportation," *IEEE Proceedings - Software Engineering*, vol. 144, no. 1, pp. 51–60, 2002.
- [9] M. Pipattanasomporn, H. Feroze, and S. Rahman, "Multi-agent systems in a distributed smart grid: Design and implementation," in *2009 IEEE/PES Power Systems Conf. and Exposition*, Mar 2009, pp. 1–8.
- [10] 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.
- [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 *57th IEEE Conf. on Decision and Control*, Dec 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 *53rd IEEE Conf. on Decision and Control*, Dec 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] J. H. Holland, "Genetic Algorithms and Adaptation," in *Adaptive Control of Ill-Defined Systems*. Boston, MA: Springer US, 1984, pp. 317–333.
- [15] 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.
- [16] X. Sun, C. G. Cassandras, and X. Meng, "Exploiting submodularity to quantify near-optimality in multi-agent coverage problems," *Automatica*, vol. 100, pp. 349–359, Feb 2019.
- [17] 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.
- [18] S. Welikala and C. G. Cassandras, "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>
- [19] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2016.
- [20] H. Flanders, "Differentiation Under the Integral Sign," *The American Mathematical Monthly*, vol. 80, no. 6, pp. 615–627, 1973.