`ARGFree`: A Randomized Gradient-Free algorithm for Aggregative Cooperative Optimization and Applications to Robotic Formation (Extended Version)

Amir Mehrnoosh, Giuseppe Speciale, Riccardo Brumali, Giuseppe Notarstefano, Gianluca Bianchin

***Abstract*— Aggregative cooperative optimization problems arise in distributed decision-making scenarios where each agent's objective depends on its own decision as well as on an aggregate variable representing the collective behavior of the system. Motivated by practical settings in which gradient information is unavailable, this paper proposes a randomized gradient-free algorithm, named `ARGFree`, for solving such problems. We establish that `ARGFree` converges in expectation to an approximate optimizer, where the approximation error originates from the use of a randomized gradient estimator. To the best of our knowledge, `ARGFree` is the first method in the literature capable of solving aggregative cooperative optimization problems without requiring gradient information. The effectiveness of the proposed algorithm is validated through robotic formation control experiments, including an implementation on a team of embedded systems based on Segway-type robots.**

I. Introduction

In recent years, the widespread use of multi-agent systems has sparked increasing interest in solving optimization problems through distributed approaches. Representative examples include parameter estimation, source localization in sensor networks, utility maximization, resource allocation, and multi-robot coordination. See the recent surveys [1]–[3] for comprehensive overviews of the field.

A large body of work on cooperative distributed optimization focuses on the so-called *consensus optimization* or *federated learning* framework [1]. In this setting, agents aim to jointly solve optimization problems of the form $\min_{x_1,\dots,x_N} \frac{1}{N}\sum_{i=1}^{N} f_i(x_i)$, subject to $x_i = x_j$ for all $i \neq j$. In this formulation, f_i denotes the local loss function that agent i seeks to minimize, and the consensus constraints $x_i = x_j$ ensures that all decision variables agree at convergence. One key feature of this problem is that each f_i depends only on the the local decision variable x_i. However, in many practical problems—such as robotic formation [4] and feedback optimization [5], [6]—local objective functions may depend not only on the agent's own decision variable, but also on those of other agents. Moreover, the decision variables may not need to always coincide at convergence (such as in k-agreement problems [7]). For example, in robotic formation control problems, each agent is interested in reaching a configuration that depends not only on the individual target location, but also on the barycenter of the group. These objectives have inspired the framework of *aggregative cooperative optimization*, recently proposed in [8], where agents cooperatively solve optimization problems of the form $\min_{x_1,\dots,x_N} \frac{1}{N}\sum_{i=1}^{N} \tilde{f}_i(x_i, \sigma_{\mathrm{f}}(x))$, with $\sigma_{\mathrm{f}}(x)$ denoting an aggregation function that depends on all decision variables $x_1, \dots, x_N$. Unfortunately, existing techniques to solve aggregative cooperative optimization problems assume that the agents have access to gradient (or subgradient) information about the local objective. In fact, in various applications, the relationship between the decision variables and the cost functions may be unknown, gradient information may be inaccessible, or the functions may not even be differentiable. The dependence on the aggregative function, along with the potentially complex structure of $\sigma_{\mathrm{f}}(x)$, further increases the difficulty of computing gradients. Motivated by this gap, we propose `ARGFree`, a distributed gradient-free method that employs randomized (zeroth-order) finite-difference approximations to estimate gradients and solve aggregative cooperative optimization problems without requiring explicit gradient information.

The `A`ggregative `R`andom `G`radient-`Free` (`ARGFree`) algorithm proposed here builds upon two main components: (i) a descent step, based on a forward-difference approximation of the gradient, which drives the optimization toward a minimizer of the aggregate loss function; and (ii) a group of tracking variables, designed to estimate finite-difference approximations of the gradient of the loss function. We demonstrate that the proposed method is capable of computing an approximate solution to the optimization problem, where the approximation error stems from using a randomized gradient estimate in place of the exact gradient.

We classify the existing literature relevant to our work into two main categories: (i) research on aggregative cooperative optimization, and (ii) gradient-free methods for distributed optimization. *Aggregative cooperative optimization:* The aggregative cooperative optimization framework was introduced in the pioneering work [8] to model problems where local objectives depend on a global (aggregative) variable. It is worth emphasizing that our use of the term "cooperative" in *aggregative cooperative optimization* is intended to distinguish our framework from that of *aggregative games* [9]. Online and constrained variants of the aggregative cooperative optimization problem have been studied in [10]. Other notable contributions include [11], which introduces a distributed Frank–Wolfe method, and the accelerated algo-

A. Mehrnoosh and G. Bianchin are with the ICTEAM institute and the Department of Mathematical Engineering (INMA) at the University of Louvain, Belgium. Email: {amir.mehrnoosh,gianluca.bianchin}@uclouvain.be R. Brumali and G. Notarstefano are with the Department of Electrical, Electronic and Information Engineering, Alma Mater Studiorum–Università di Bologna, Bologna, Italy. Email: {riccardo.brumali,giuseppe.notarstefano}@unibo.it.

rithms proposed in [12]. Particularly relevant to the present work are the recent works [13], [14], which harness learning-based techniques to handle uncertainty in the environment.

Gradient-free methods in distributed optimization: Although gradient-free techniques have a long history in optimization, their theoretical analysis was formalized only recently in [15] in the centralized setting. In distributed settings, most existing works focus on the consensus optimization problem. Methods based on multi-point gradient estimators are studied in [16], two-point estimators are proposed in [17], and single-point estimators are considered in [18]. Continuous-time algorithms have also been shown to be effective [19]. Further developments include analysis over time-varying graphs [20], primal-dual approaches [21], constrained stochastic problems [22], and communication-imperfect settings [23]. Accelerated variants are proposed in [24], while extremum-seeking based approaches are studied in [25]. Gradient-free methods tailored for games are presented in [26]. To the best of the authors' knowledge, all these methods are limited to the consensus optimization framework, and gradient-free methods specifically designed for aggregative cooperative optimization problems are still lacking.

This paper features three main contributions. First, we introduce `ARGFree` (Algorithm 1), a randomized, gradient-free method designed to solve aggregative cooperative optimization problems. Unlike existing approaches, our method does not require gradient information; instead, it relies only on local function evaluations (as detailed in requirements (R1)–(R2) in Section II). Second, we establish convergence bounds for the proposed algorithm (theorems 7 and 8), showing that its iterates converge in expectation to an approximate optimizer; the approximation error arises from the use of a randomized gradient estimator in place of the exact gradient. An important feature of the algorithm is that its asymptotic accuracy can be controlled by appropriately tuning the available parameters. Third, we validate our theoretical findings through both numerical simulations and experiments using a team of embedded systems based on the Balboa 32U4 self-balancing robot (Section V).

The remainder of the paper is organized as follows. Section II introduces the problem formulation and technical preliminaries. Section III presents the proposed `ARGFree` algorithm and analyzes its convergence properties. Section V provides numerical and experimental validations on robotic formation control tasks. Finally, Section VI concludes the paper.

We adopt the following notation. $\mathbb{R}^n_{>0}$ and $\mathbb{R}^n_{\geq 0}$ denote vectors in $\mathbb{R}^n$ with positive and nonnegative entries, respectively; $\mathbb{S}^n$ is the space of symmetric $n \times n$ real matrices; and $[n] := \{1, \dots, n\}$. For vectors $v_1, \dots, v_n$, $\mathrm{col}(v_1, \dots, v_n)$ denotes their column stacking. $\mathbb{1}_n$ and $\mathbb{0}_n$ denote the n-dimensional all-ones and all-zeros vectors, respectively (dimensions are omitted when clear). For a square matrix M, $\rho(M)$ denotes its spectral radius, and $\otimes$ the Kronecker product. $E_u[\cdot]$ denotes expectation with respect to the random variable u.

II. Problem setting

In this section, we introduce the problem of interest and illustrate its relevance through representative applications.

A. Problem formulation

Consider a group of N agents, where each agent $i \in [N]$ is associated with a local decision variable $x_i \in \mathbb{R}^{n_i}$ and a local loss function $\tilde{f}_i : \mathbb{R}^{n_i} \times \mathbb{R}^d \to \mathbb{R}$. Let $x = \mathrm{col}(x_1, \dots, x_N)$ and define $n := \sum_{i=1}^N n_i$. We assume that each loss function $\tilde{f}_i$ depends not only on the local decision x_i but also on a global quantity $\sigma_{\mathrm{f}}(x) \in \mathbb{R}^d$, referred to as the *aggregative variable*, which aggregates information from all agents as

$$\sigma_{\mathrm{f}}(x) := \frac{1}{N} \sum_{i=1}^{N} \phi_i(x_i), \tag{1}$$

where $\phi_i : \mathbb{R}^{n_i} \to \mathbb{R}^d$. The agents aim to collaboratively solve the following optimization[1] problem:

$$\min_{x \in \mathbb{R}^n} f(x) := \frac{1}{N} \sum_{i=1}^{N} \tilde{f}_i(x_i, \sigma_{\mathrm{f}}(x)). \tag{2}$$

Equations (1)–(2) formalize an *aggregative cooperative optimization problem* [8], in which the agents aim to minimize a global objective defined as the average of their local costs. It is worth noting that, unlike in aggregative games [9], where $\sigma_{\mathrm{f}}(x)$ is typically assumed to be independent of x_i, problem (2) explicitly accounts for this dependence.

In this work, we focus on scenarios in which the agents operate under the following Requirements (R):

(R1) The functions $\tilde{f}_i(\cdot, \cdot)$ and $\phi_i(\cdot)$, along with the decision variable x_i, need to be kept private to agent i and, therefore, are not known by any other agent $j \neq i$.

(R2) Agent i does not have access to the analytic expressions (nor the derivatives) of $\tilde{f}_i(\cdot, \cdot)$ or $\phi_i(\cdot)$; instead, these functions can only be evaluated through oracle queries:

$$(x_i, \sigma) \mapsto \tilde{f}_i(x_i, \sigma), \qquad x_i \mapsto \phi_i(x_i),$$

with $x_i \in \mathbb{R}^{n_i}, \sigma \in \mathbb{R}^d$.

Requirements (R1)–(R2) naturally occur in many practical scenarios; a representative example is the robotic formation control problem discussed in Section V. Solving (2) under (R1)–(R2) is challenging for two main reasons: (i) due to (R1), centralized algorithms are inapplicable, necessitating a distributed architecture in which each agent i relies only on its local functions $\tilde{f}_i(\cdot, \cdot)$ and $\phi_i(\cdot)$; and (ii) due to (R2), existing methods (e.g., those in [8]) relying on gradient-descent-type iterations, are inapplicable.

Motivated by these requirements, we consider scenarios in which agents aim to solve problem (2) in a distributed manner, using only local communication and cooperative coordination. To this end, we assume that the agents can

[1] For notational clarity, throughout this work, the tilde notation (e.g., $\tilde{f}$) is used to denote functions that explicitly depend on the aggregative variable (i.e., functions of two arguments), whereas functions without a tilde (e.g., f) represent the corresponding composite (single-argument) functions. The two are related through (10), as detailed shortly below.

exchange information with their neighbors; we model the communication topology using a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the node set $\mathcal{V} = \{1, \dots, N\}$ models the agents and the edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ describes the communication links. We impose the following assumption on $\mathcal{G}$ (see Section II-B for the adopted graph-theoretic notation).

Assumption 1 (***Properties of the communication graph***)**.** The digraph $\mathcal{G}$ is strongly connected. Moreover, $\mathcal{G}$ admits an adjacency matrix A that is doubly stochastic. □

Assumption 1 is standard in the design of coordination schemes and distributed algorithms (e.g., consensus averaging) [27]–[29]. Intuitively, the strong connectivity condition guarantees that information can (asymptotically) propagate from any node of $\mathcal{G}$ to every other node (note that it does not necessarily require that the graph is complete [27]). The doubly stochasticity requirement is, for instance, automatically satisfied when $\mathcal{G}$ is additionally aperiodic or when each node possesses a self-loop [30]. We also note that several procedures are available to construct matrices A that satisfy this assumption; see [27] for centralized methods and [30] for distributed ones. In the remainder, we let $A \in \mathbb{R}^{N\times N}$ be a matrix as in Assumption 1.

We now formally state the problem studied in this work.

Problem 1 (***Objective of this work***)**.** Design a distributed algorithm, compatible with the graph topology $\mathcal{G}$, enabling the agents to cooperatively compute solutions to problem (2), subject to requirements (R1)–(R2). □

B. Preliminaries

We present hereafter basic properties used throughout the paper.

a) Basic notions on algebraic graph theory: For a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we adopt the convention that an edge $(j, i) \in \mathcal{E}$ indicates that node j is able to receive information from i (or, equivalently, i transmits information to j). For a node $i \in \mathcal{V}$, we denote by $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ the set of agents that send information to i. Matrix $A = [a_{ij}] \in \mathbb{R}^{N\times N}$ is said to be an *adjacency matrix* for $\mathcal{G}$ if it satisfies $a_{ij} > 0$ if $(j, i) \in \mathcal{E}$, and $a_{ij} = 0$ otherwise. A is said to be *doubly stochastic* if $\sum_{j=1}^{N} a_{ij} = 1$ and $\sum_{i=1}^{N} a_{ij} = 1$. We let $J := \frac{1}{N}\mathbb{1}_N\mathbb{1}_N^\top \in \mathbb{R}^{N\times N}$ and $\mathcal{J} := J \otimes I_d \in \mathbb{R}^{Nd\times Nd}$. Notice that $\|\mathcal{J} - I\| = 1$. With a slight abuse of notation, we denote by $\rho_A = \|A - J\|$ the operator norm of A (which, in general, differs from its spectral radius $\rho(A)$). Let $v = \mathrm{col}(v_1, \dots, v_N) \in \mathbb{R}^{Nd}$ with $v_i \in \mathbb{R}^d, i \in [N]$, and recall that $\bar{v} := \frac{1}{N}\sum_{i=1}^{N} v_i \in \mathbb{R}^d$ denotes the entries average. Recalling the well-known [31] property $(A \otimes B)(C \otimes D) = AC \otimes BD$ for A, B, C, D of suitable dimensions, we have

$$\mathbb{1}_N \otimes \bar{v} = \mathcal{J}v. \tag{3}$$

Given $A \in \mathbb{R}^{N\times N}$, we denote by $\mathcal{A} := A \otimes I_d \in \mathbb{R}^{Nd}$. The following lemma is instrumental to our analysis.

Lemma 1 ([31])**.** Let $A \in \mathbb{R}^{N\times N}$ be a doubly stochastic matrix. Then, ρ_A satisfies $\rho_A < 1$. Moreover,

$$\mathcal{A}\mathcal{J} = \mathcal{J}\mathcal{A} = \mathcal{J}, \quad \|\mathcal{A}x - \mathcal{J}x\| \le \rho_A\|x - \mathcal{J}x\|, \tag{4}$$

for any $x \in \mathbb{R}^{Nd}$. □

b) Notions on Gaussian smoothing approximations: Consider a function $f : \mathbb{R}^n \to \mathbb{R}$ and assume that, at each point $x \in \mathbb{R}^n$, it is differentiable along any direction. The *Gaussian approximation* of $f(x)$ is defined as:

$$f_\delta(x) = \frac{1}{\kappa}\int_{\mathbb{R}^n} f(x + \delta u)\mathrm{e}^{-\frac{1}{2}\|u\|^2}\,\mathrm{d}u, \tag{5}$$

where $\delta \in \mathbb{R}_{>0}$ and $\kappa := \int_{\mathbb{R}^n} \mathrm{e}^{-\frac{1}{2}\|u\|^2}\,\mathrm{d}u$. Observe that, if we let $u \sim \mathcal{N}(0, \Sigma)$, then $f_\delta(x) \equiv E_u[f(x + \delta u)]$; in this case, $\kappa = (2\pi)^{n/2}(\det\Sigma)^{1/2}$. Given $u \sim \mathcal{N}(0, \Sigma)$, we define the *forward-difference gradient-free oracle* as follows:

$$g_\delta(x) = \frac{f(x + \delta u) - f(x)}{\delta}\Sigma^{-1}u, \tag{6}$$

where $\delta \in \mathbb{R}_{>0}$. Intuitively, in (6), the vector u can be interpreted as a random perturbation or directional probe that excites the function $f(x)$ to estimate its gradient. The following results are instrumental to our analysis.

Lemma 2 ([15, Lem. 1])**.** Let $p \in \mathbb{Z}_{\ge 0}$ and $M_p := \frac{1}{\kappa}\int_{\mathbb{R}^n} \|u\|^p\mathrm{e}^{-\frac{1}{2}\|u\|^2}\,\mathrm{d}u$. Then, $M_0 = 1, M_1 = n$, and

$$M_p \le \begin{cases} n^{p/2}, & \text{if } p \in [0, 2], \\ (p + n)^{p/2}, & \text{if } p > 2. \end{cases} \tag{7}$$

□

Lemma 3 ([15, Thm. 2])**.** Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is convex and Lipschitz continuous. Then,

$$E_u[g_\delta(x)] = \nabla f_\delta(x) \tag{8}$$

for any $x \in \mathbb{R}^n$ □

Lemma 4 ([15, Thm.s 1 and 4, Lemmas 4 and 5])**.** Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is convex and L_1-smooth. Then, for any $x \in \mathbb{R}^n$,

$$|f_\delta(x) - f(x)| \le \frac{\delta^2}{2}L_1 n, \tag{9a}$$

$$f_\delta(x) - f(x) \ge 0, \tag{9b}$$

$$\|\nabla f(x)\|^2 \le 2\|\nabla f_\delta(x)\|^2 + \frac{\delta^2}{2}L_1^2(n + 6)^3, \tag{9c}$$

$$E_u[\|g_\delta(x)\|^2] \le \frac{\delta^2}{2}(n + 6)^3L_1^2 + 2(n + 4)\|\nabla f(x)\|^2, \tag{9d}$$

$$E_u[\|g_\delta(x)\|^2] \le 4(n + 4)\|\nabla f_\delta(x)\|^2 + 3\delta^2L_1^2(n + 4)^3. \tag{9e}$$

□

III. Algorithm design and convergence guarantees

In this section, we introduce an iterative algorithm for solving Problem 1 and establish its convergence guarantees.

A. The `ARGFree` *algorithm*

The proposed method, called `A`ggregative `R`andom `G`radient-`Free` (`ARGFree`) algorithm, is presented in Algorithm 1. The algorithm is structured as follows: each agent $i \in [N]$, updates its local decision variable x_i^k using a forward-difference optimization scheme (see line 1), driven by the

Algorithm 1: `ARGFree` (agent i)

Data: Parameters $\alpha, \delta \in \mathbb{R}_{>0}$,
Initializations: Set $k = 0, u_i^0 \sim \mathcal{N}(0, I_{n_i})$, $x_i^0 \in \mathbb{R}^{n_i}$,

$$\sigma_i^0 = \phi_i(x_i^0), \qquad s_i^0 = \phi_i(x_i^0 + \delta u_i^0),$$
$$z_i^0 = \tilde{f}_i(x_i^0, \sigma_i^0) \qquad p_i^0 = \tilde{f}_i(x_i^0 + \delta u_i^0, s_i^0)$$

Optimization variable update:

1
$$x_i^{k+1} = x_i^k - \alpha \frac{p_i^k - z_i^k}{\delta} u_i^k$$

Tracking variables update:

2 Generate $u_i^{k+1} \sim \mathcal{N}(0, I_{n_i})$ and update:

3 $$\sigma_i^{k+1} = \sum_{j \in \mathcal{N}_i} a_{ij} \sigma_j^k + \phi_i(x_i^{k+1}) - \phi_i(x_i^k)$$

4 $$s_i^{k+1} = \sum_{j \in \mathcal{N}_i} a_{ij} s_j^k + \phi_i(x_i^{k+1} + \delta u_i^{k+1}) - \phi_i(x_i^k + \delta u_i^k)$$

5 $$z_i^{k+1} = \sum_{j \in \mathcal{N}_i} a_{ij} z_j^k + \tilde{f}_i(x_i^{k+1}, \sigma_i^{k+1}) - \tilde{f}_i(x_i^k, \sigma_i^k)$$

6 $$p_i^{k+1} = \sum_{j \in \mathcal{N}_i} a_{ij} p_j^k + \tilde{f}_i(x_i^{k+1} + \delta u_i^{k+1}, s_i^{k+1}) - \tilde{f}_i(x_i^k + \delta u_i^k, s_i^k)$$

7 Transmit σ_i^{k+1}, s_i^{k+1}, z_i^{k+1} p_i^{k+1} to neighbors
8 Set $k \leftarrow k + 1$ and go to line 1
Result: x_i^k, estimate for the optimizer of (2)

term $\alpha \frac{p_i^k - z_i^k}{\delta} \cdot u_i^k$, where $\alpha > 0$ is the algorithm's stepsize and $\delta > 0$ is a tunable parameter (hereafter called *smoothing ratio*—see (5)). In this operation, the vector $u_i^k \in \mathbb{R}^{n_i}$ describes a random perturbation direction, while the variables p_i^k and z_i^k model local estimates for, respectively, the quantities $f(x_k + \delta u_k)$ and $f(x_k)$. To update the estimates p_i^k and z_i^k, the algorithm uses two dynamic consensus tracking schemes (see lines 5 and 6) driven, locally, by the signals $\tilde{f}_i(x_i^k + \delta u_i^k, \sigma_{\mathrm{f}}(x^k + \delta u^k))$ and $\tilde{f}_i(x_i^k, \sigma_{\mathrm{f}}(x^k))$, respectively. Finally, because evaluating the quantities $\sigma_{\mathrm{f}}(x^k + \delta u^k)$ and $\sigma_{\mathrm{f}}(x^k)$ would require global knowledge (through knowledge of the global vectors x^k, u^k as well as of the functions $\phi_j(\cdot)$), these quantities are replaced by the local estimates s_i^k and σ_i^k in lines 5–6. In other words, s_i^k and σ_i^k are interpreted as local proxies for, respectively, $\sigma_{\mathrm{f}}(x^k + \delta u^k)$ and $\sigma_{\mathrm{f}}(x^k)$, and estimated through a dynamic consensus tracking scheme (see lines 3–4).

B. Convergence guarantees for `ARGFree`

We now establish convergence guarantees for `ARGFree`. Before proceeding, we introduce some notation that will be instrumental for stating the main results. Define $\tilde{f} : \mathbb{R}^n \times \mathbb{R}^{Nd} \to \mathbb{R}$ as

$$\tilde{f}(x, \sigma) := \frac{1}{N} \sum_{i=1}^{N} \tilde{f}_i(x_i, \sigma_i),$$

where, for a vector $\sigma \in \mathbb{R}^{Nd}$, we used the notation $\sigma = \mathrm{col}(\sigma_1, \dots, \sigma_N), \sigma_i \in \mathbb{R}^d$. Observe that, by (1)–(2), the following identity holds

$$\tilde{f}(x, \mathbb{1} \otimes \sigma_{\mathrm{f}}(x)) = f(x). \tag{10}$$

Next, we introduce the vector notation:

$$\begin{aligned} x_k &= \mathrm{col}\left(x_1^k, \dots, x_N^k\right) \in \mathbb{R}^n, & u_k &= \mathrm{col}\left(u_1^k, \dots, u_N^k\right) \in \mathbb{R}^n, \\ \sigma_k &= \mathrm{col}\left(\sigma_1^k, \dots, \sigma_N^k\right) \in \mathbb{R}^{Nd}, & s_k &= \mathrm{col}\left(s_1^k, \dots, s_N^k\right) \in \mathbb{R}^{Nd}, \\ z_k &= \mathrm{col}\left(z_1^k, \dots, z_N^k\right) \in \mathbb{R}^N, & p_k &= \mathrm{col}\left(p_1^k, \dots, p_N^k\right) \in \mathbb{R}^N. \end{aligned} \tag{11}$$

Using this notation the updates of Algorithm 1 can be written in vector form[2] as:

$$x_{k+1} = x_k - \frac{\alpha}{\delta}(p_k - z_k) \odot u_k, \tag{12a}$$
$$\sigma_{k+1} = \mathcal{A}\sigma_k + \phi_{\mathrm{v}}(x_{k+1}) - \phi_{\mathrm{v}}(x_k), \tag{12b}$$
$$s_{k+1} = \mathcal{A}s_k + \phi_{\mathrm{v}}(x_{k+1} + \delta u_{k+1}) - \phi_{\mathrm{v}}(x_k + \delta u_k), \tag{12c}$$
$$z_{k+1} = \mathcal{A}z_k + \tilde{f}_{\mathrm{v}}(x_{k+1}, \sigma_{k+1}) - \tilde{f}_{\mathrm{v}}(x_k, \sigma_k), \tag{12d}$$
$$p_{k+1} = \mathcal{A}p_k + \tilde{f}_{\mathrm{v}}(x_{k+1} + \delta u_{k+1}, s_{k+1}) - \tilde{f}_{\mathrm{v}}(x_k + \delta u_k, s_k), \tag{12e}$$

where $\phi_{\mathrm{v}} : \mathbb{R}^n \to \mathbb{R}^{Nd}$ and $\tilde{f}_{\mathrm{v}} : \mathbb{R}^n \times \mathbb{R}^{Nd} \to \mathbb{R}^N$ are:

$$\begin{aligned} \phi_{\mathrm{v}}(x) &:= \mathrm{col}\left(\phi_1(x_1), \dots, \phi_N(x_N)\right), \\ \tilde{f}_{\mathrm{v}}(x, \sigma) &:= \mathrm{col}(\tilde{f}_1(x_1, \sigma_1), \dots, \tilde{f}_N(x_N, \sigma_N)). \end{aligned} \tag{13}$$

We are now ready to present our first result, which formalizes the tracking properties of the dynamic consensus tracking scheme used in lines 3–6 of Algorithm 1. Recall that, for $v = \mathrm{col}(v_1, \dots, v_N)$, the notation $\bar{v} := \frac{1}{N}\sum_{i=1}^N v_i$ denotes the average of its elements (see Section I for notation).

Lemma 5 (***Properties of the tracking variables***)**.** Suppose Assumptions 1–2 hold. Then, the states $(\sigma_k, s_k, z_k, p_k)$ generated by Algorithm 1 satisfy:

$$\begin{aligned} \bar{\sigma}_k &= \sigma_{\mathrm{f}}(x_k), & \bar{s}_k &= \sigma_{\mathrm{f}}(x_k + \delta u_k), \\ \bar{z}_k &= \tilde{f}(x_k, \sigma_k), & \bar{p}_k &= \tilde{f}(x_k + \delta u_k, \sigma_k), \end{aligned} \tag{14}$$

at every $k \in \mathbb{Z}_{\geq 0}$. □

The proof of this claim is presented in the appendix. Lemma 5 establishes that the averages (computed across the agents) of the state variables σ_k, s_k, z_k, p_k coincide, respectively, with the quantities $\sigma_{\mathrm{f}}(x_k)$, $\sigma_{\mathrm{f}}(x_k + \delta u_k)$, $\tilde{f}(x_k, \sigma_k)$, and $\tilde{f}(x_k + \delta u_k, \sigma_k)$. This result implies that the averages across the network of the algorithm's state variables σ_k, s_k, z_k, p_k track, respectively, the quantities they are designed to represent (see the interpretation of the algorithm's variables presented in Section III-A).

[2]Given two vectors, $v = \mathrm{col}(v_1, \dots, v_N) \in \mathbb{R}^N, v_i \in \mathbb{R}$, and $u = \mathrm{col}(u_1, \dots, u_N) \in \mathbb{R}^n, u_i \in \mathbb{R}^{n_i}$, we denote by $v \odot u = (v_1 u_1, \dots, v_N u_N) \in \mathbb{R}^n$ their entrywise product.

Next, we state an instrumental lemma that will be used to establish the convergence properties of Algorithm 1.

Lemma 6 (***Contraction of randomized descent under forward-difference gradient approximation***)**.** Let $f:\mathbb{R}^n\to\mathbb{R}$ be an L_1-smooth and μ-strongly convex function, and let x^* be such that $\nabla f(x^*)=0$. Let $u\sim\mathcal{N}(0,\Sigma)$ and consider the forward-difference gradient-free oracle from (6):

$$g_\delta(x)=\frac{f(x+\delta u)-f(x)}{\delta}\Sigma^{-1}u.$$

Then, for any $x\in\mathbb{R}^n$, the following inequality holds:

$$E_u[\|x-\alpha g_\delta(x)-x^*\|]\leq\sqrt{1-\beta_1^\alpha}\|x-x^*\|+\beta_2^\alpha, \quad (15)$$

where

$$\beta_1^\alpha:=\alpha\mu(1-2\alpha(n+4)L_1), \quad (16)$$
$$\beta_2^\alpha:=\sqrt{\alpha\delta^2L_1(n+\frac{\alpha}{2}(n+6)^3L_1)}. \qquad \square$$

The proof of this claim is presented in the appendix. Lemma 6 considers the iteration $x_{k+1}=x_k-\alpha g_\delta(x_k)$, which can be interpreted as a (centralized) gradient-descent-type method that employs the forward-difference approximation (6) in place of the exact gradient. The estimate (15) establishes that these iterates are contractive with respect to the optimizer x^*. Specifically, the contraction occurs with rate $\sqrt{1-\beta_1^\alpha}$, up to a neighborhood of radius β_2^α. Here, convergence to an inexact point arises from employing a randomized gradient estimate instead of the exact gradient. The convergence rate and accuracy of the algorithm depend on several optimization parameters; in particular, note that $\beta_1^\alpha<1$ requires a stepize $\alpha<\frac{1}{2(n+4)L_1}$, and that the radius of the convergence neighborhood (i.e., β_2^α) can be controlled (i.e., made arbitrarily small) by reducing the smoothing ratio δ.

Motivated by the statement of Lemma 6, we impose the following requirements on the optimization problem (2).

Assumption 2 (***Properties of the loss functions***)**.** The following statements hold:

- (A2a) The function $f(x)$ is Lipschitz smooth and μ-strongly convex. We denote by $L_0,L_1>0$ constants such that $\|f(x)-f(x')\|\leq L_0\|x-x'\|$ and $\|\nabla f(x)-\nabla f(x')\|\leq L_1\|x-x'\|$, $\forall x,x'\in\mathbb{R}^n$.
- (A2b) For each $i\in[N]$, the function $\phi_i(x_i)$ is Lipschitz continuous. We let $L_\phi>0$ be such that $\|\phi_i(x_i)-\phi_i(x_i')\|\leq L_\phi\|x_i-x_i'\|$, $\forall x_i,x_i'\in\mathbb{R}^{n_i}$, $i\in[N]$.
- (A2c) For each $i\in[N]$, the function $\tilde{f}_i$ is Lipschitz continuous. We let $\tilde{L}_{0,i}>0$ be such that

$$\|\tilde{f}_i(x_i,\sigma)-\tilde{f}_i(x_i',\sigma')\|\leq\tilde{L}_{0,i}\|\begin{bmatrix}x_i\\\sigma\end{bmatrix}-\begin{bmatrix}x_i'\\\sigma'\end{bmatrix}\|,$$

$\forall x_i,x_i'\in\mathbb{R}^{n_i},\sigma,\sigma'\in\mathbb{R}^d$. Moreover, we let $\tilde{L}_0:=\max_i\tilde{L}_{0,i}$. $\square$

Note that strong convexity is required only for the global objective $f(\cdot)$, while the individual local objectives $\tilde{f}_i(\cdot,\cdot)$ may possibly be non-convex or non-smooth. Additionally, note that no differentiability is assumed for $\phi_i(\cdot)$.

We are now ready to present a convergence estimate for Algorithm 1, which is the main result of this paper. To this end, we first introduce the following instrumental notation:

$$\theta_k:=\mathrm{col}(\|x_k-x^*\|,\|\sigma_k-\mathcal{J}\sigma_k\|,\|s_k-\mathcal{J}s_k\|, \quad (17)$$
$$\|z_k-\mathcal{J}z_k\|,\|p_k-\mathcal{J}p_k\|),$$

where $x^*\in\mathbb{R}^n$ is the unique (see (A2a)) optimizer of (2).

Theorem 7 (***Convergence estimate for Algorithm 1***)**.** Let Assumptions 1-2 hold, suppose $\delta<\alpha\sqrt{n}$, $\tilde{L}_0<\frac{1-\rho_A}{\|A-I\|}$, and that the stepsize α satisfies

$$0<\alpha<\min\left\{\frac{1}{2(n+4)L_1},\alpha_1^*,\alpha_2^*\right\}, \quad (18)$$

where

$$\alpha_1^*:=\frac{1-\rho_A}{L_\phi\left(\sqrt{2(n+4)}L_1+\frac{2\sqrt{n}}{\delta}\right)},$$
$$\alpha_2^*:=\frac{1-\rho_A-\tilde{L}_0\|A-I\|}{\tilde{L}_0\left(1+L_\phi\right)\left(\frac{2\sqrt{n}}{\delta}+\sqrt{2(n+4)}L_1\right)}.$$

Then, there exists $\eta\in(0,1)$ and $\varepsilon\in\mathbb{R}_{>0}$ such that, for all $k\in\mathbb{Z}_{\geq0}$, the iterates of Algorithm 1 satisfy:

$$E[\theta_k]\leq\eta^kE[\theta_0]+\frac{1-\eta^k}{1-\eta}\varepsilon. \quad (19)$$

Moreover[3], for large[4] n,

$$\varepsilon=\mathcal{O}(\delta\cdot E[\|u_{k+1}-u_k\|^2])=\mathcal{O}(\delta n). \quad (20)$$

$\square$

The proof of this claim is presented in Section IV. Theorem (7) establishes that the iterates of Algorithm 1 converge at rate η to a neighborhood of the solution of (2). Note that precise estimates for η and ϵ are given shortly below (see Theorem 8). In analogy with (15), convergence to an inexact point arises from employing a randomized gradient estimate instead of the exact gradient. The algorithm's accuracy ε depends on the smoothing ratio δ and on the problem dimension n, and can be controlled (i.e., made arbitrarily small) by reducing δ. The algorithm's stepsize α is required to be sufficiently small, as given by the estimate (18). Note that the upper bound $\alpha<\frac{1}{2(n+4)L_1}$ recovers the maximum stepsize allowed for the centralized method (see Lemma 6). The additional bounds $\alpha<\alpha_1^*$ and $\alpha<\alpha_2^*$ are required to ensure contraction of the dynamic consensus tracking variables (lines 3–6 of Algorithm 1). Notice also that α_2^* is guaranteed to be a real positive number under the requirement $\tilde{L}_0<\frac{1-\rho_A}{\|A-I\|}$, which can be interpreted as an upper bound on the largest admitted variability on the functions $\tilde{f}_i$ (cf. (A2c)) in relation to the averaging rate of the communication graph (measured by the parameters ρ_A and $\|A-I\|$). Intuitively, the larger $\tilde{L}_0$, the faster the communication graph needs to be at averaging (cf. Assumption 1). We conclude by giving precise estimates for η and ε next.

[3]We say that $f(x)=\mathcal{O}(g(x))$ as $x\to\infty$ if there exist constants $C>0$ and $x_0\in\mathbb{R}$ such that: $|f(x)|\leq C\cdot|g(x)|$ for all $x\geq x_0$.

[4]That is, there exists $n_\circ$, such that, for any $n\geq n_\circ$, the estimate holds.

Theorem 8 (***Estimates for the convergence rate and accuracy of Algorithm 1***)**.** Under the assumptions of Theorem 7, the convergence rate in (19) satisfies:

$$\eta \leq \max\{\eta_1^*, \eta_2^*, \eta_3^*\}, \tag{21}$$

where

$$\eta_1^* := \sqrt{1-\alpha\mu\left(1-2\alpha(n+4)L_1\right)} + \frac{2\alpha\sqrt{n}}{\delta},$$
$$\eta_2^* := \rho_A + \alpha L_\phi\left(\sqrt{2(n+4)}L_1 + \frac{2\sqrt{n}}{\delta}\right),$$
$$\eta_3^* := \rho_A + \alpha\tilde{L}_0\left(1+L_\phi\right)\left(\sqrt{2(n+4)}L_1 + \frac{2\sqrt{n}}{\delta}\right) + \tilde{L}_0\|A-I\|.$$

Moreover, the algorithm's accuracy ε satisfies:

$$\varepsilon = \delta\left[\alpha L_1 n + 2L_{\text{comb}}^2 E[\|u_{k+1}-u_k\|^2]^2 + \frac{\alpha^2 L_1^2}{2}(n+6)^3\left(1+\frac{3}{2}L_{\text{comb}}^2\right)\right]^{1/2}, \tag{22}$$

where $L_{\text{comb}}^2 := L_\phi^2 + \tilde{L}_0^2\left(1+L_\phi\right)^2$. □

The proof of this claim is presented in Section IV. It is worth comparing the convergence rate estimate established by Theorem 8:

$$\eta_1^* = \sqrt{1-\alpha\mu\left(1-2\alpha(n+4)L_1\right)} + \frac{2n\alpha^2}{\delta^2},$$

with the corresponding estimate for the centralized algorithm given in Lemma 6 (see (15)):

$$\sqrt{1-\beta_1^\alpha} = \sqrt{1-\alpha\mu\left(1-2\alpha(n+4)L_1\right)}.$$

The degradation in rate affecting the distributed algorithm (characterized by the additional term $\frac{2n\alpha^2}{\delta^2}$) can be traced back to the presence of the tracking variables σ_k, s_k, z_k, p_k. Intuitively, while the centralized algorithm has direct access to the quantities $f(x+\delta u)$ and $f(x)$, the distributed algorithm must rely on approximations of these quantities via the variables p_k and z_k, which in turn slows down the descent step. The remaining estimates, η_2^* and η_3^*, can be interpreted as bounds on the convergence rates of the tracking variables—specifically, η_2^* corresponds to the convergence of (σ_k, s_k), while η_3^* pertains to (z_k, p_k).

IV. Convergence Analysis of the Algorithms

This section is devoted to establishing the bounds presented theorems 7 and 8. Our approach is based on showing that, for all $k \in \mathbb{Z}_{\geq 0}$, the quantity θ_k satisfies the following bound:

$$E[\theta_{k+1}] \leq M(\alpha)E[\theta_k] + b, \tag{23}$$

where $M(\alpha)$ is a Schur stable matrix with spectral radius η and b is such that $\|b\| \leq \varepsilon$. We begin with the proof of Theorem 7, which is organized into seven subsections (Section IV-A–IV-G). We conclude by presenting the proof of Theorem 8 in Section IV-H.

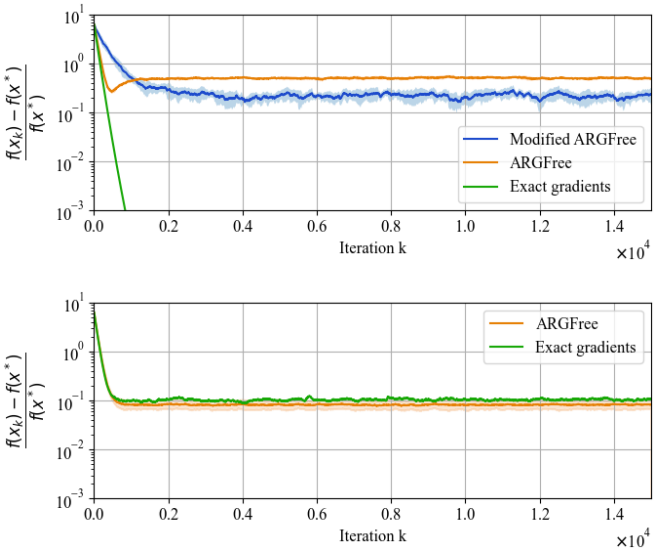


Fig. 1. Simulation results comparing the proposed methods with gradient-based techniques from [8] for a robotic formation control problem. All results are averaged over 10 Monte Carlo runs, and shaded regions represent $\pm\sigma$ confidence intervals. **(Top)** Noiseless setting. **(Bottom)** Noisy setting. The results show that, while in the absence of noise gradient-based methods achieve faster convergence and higher accuracy, under noisy conditions our approach outperforms them, despite requiring less information. See Section V-A for further discussion.

A. Bound for $E[\|x_{k+1}-x^*\|]$

We have the following estimate:

$$E_{u_k}[\|x_{k+1}-x^*\|] = E_{u_k}[\|x_k - \frac{\alpha}{\delta}(p_k-z_k)u_k - x^*\|] \leq \underbrace{E_{u_k}[\|x_k-\alpha g_\delta(x_k)-x^*\|]}_{:=ⓐ} + \underbrace{\alpha E[\|g_\delta(x_k) - \frac{p_k-z_k}{\delta}u_k\|]}_{:=ⓑ}.$$

By application of Lemma 6, we have $ⓐ \leq \sqrt{1-\beta_1^\alpha}\|x_k - x^*\| + \beta_2^\alpha$; The term ⓑ satisfies:

$$ⓑ \leq \underbrace{\alpha E_{u_k}[\|g_\delta(x_k) - \mathbb{1}\otimes\frac{\bar{p}_k-\bar{z}_k}{\delta}u_k\|]}_{:=ⓒ} + \underbrace{\frac{\alpha}{\delta}E_{u_k}[\|(p_k-\mathbb{1}\otimes\bar{p}_k)u_k\|]}_{:=ⓓ} + \underbrace{\frac{\alpha}{\delta}E_{u_k}[\|(z_k-\mathbb{1}\otimes\bar{z}_k)u_k\|]}_{:=ⓔ}, \tag{24}$$

where the first inequality follows by adding and subtracting $\mathbb{1}\otimes\frac{\bar{p}_k-\bar{z}_k}{\delta}u_k$ inside the norm. Next, observe that $ⓒ \overset{(14),(6)}{=} 0$; the remaining terms satisfy:

$$ⓓ \overset{(3),(7)}{\leq} \frac{\alpha\sqrt{n}}{\delta}\|p_k - \mathcal{J}p_k\|,$$
$$ⓔ \overset{(3),(7)}{\leq} \frac{\alpha\sqrt{n}}{\delta}\|z_k - \mathcal{J}z_k\|. \tag{25}$$

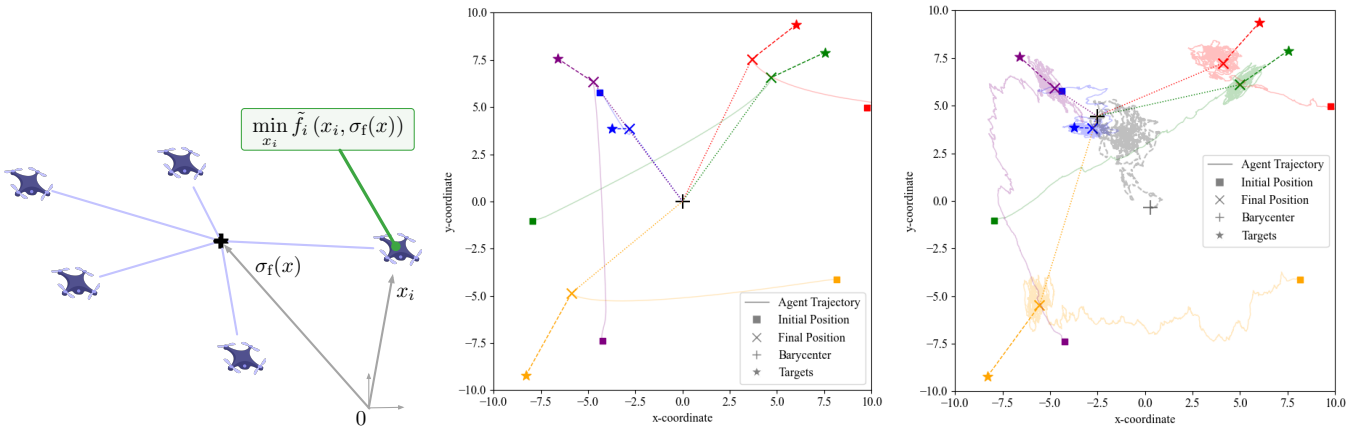


Fig. 2. **(Left)** Illustration of the robotic formation control problem studied in Section V. **(Center)** Trajectories followed by the robots when applying the algorithm from [8], which relies on exact gradient information. **(Right)** Implementation of the modified `ARGFree` algorithm. Overall, `ARGFree` achieves performance comparable to that of exact gradient methods, while requiring no gradient information for its implementation. See Section V-A for details.

Summarizing, we have derived the estimate:

$$E_{u_k}[\|x_{k+1} - x^*\|] \leq \sqrt{1-\beta_1^\alpha}\|x_k - x^*\| + \beta_2^\alpha + \gamma_1^\alpha(\|z_k - \mathcal{J}z_k\| + \|p_k - \mathcal{J}p_k\|), \tag{26}$$

where we defined $\gamma_1^\alpha =: \alpha\sqrt{n}/\delta$.

B. An auxiliary bound for $E[\|x_{k+1} - x_k\|]$

We have the following estimate:

$$\begin{aligned} E_{u_k}[\|x_{k+1} - x_k\|] &= \alpha E[\|\frac{p_k - z_k}{\delta}u_k\|] \\ &\leq \underbrace{\alpha E_{u_k}[\|g_\delta(x_k) - \frac{p_k - z_k}{\delta}u_k\|]}_{=\textcircled{b}} + \alpha E_{u_k}[\|g_\delta(x_k)\|] \\ &\overset{(24),(25)}{\leq} \gamma_1^\alpha(\|p_k - \mathcal{J}p_k\| + \|z_k - \mathcal{J}z_k\|) + \alpha E_{u_k}[\|g_\delta(x_k)\|] \\ &\overset{(9d)}{\leq} \gamma_1^\alpha(\|p_k - \mathcal{J}p_k\| + \|z_k - \mathcal{J}z_k\|) + \alpha\left(\frac{\delta^2}{2}L_1^2(n+6)^3 + 2(n+4)\|\nabla f(x)\|^2\right)^{1/2} \\ &\leq \gamma_1^\alpha(\|p_k - \mathcal{J}p_k\| + \|z_k - \mathcal{J}z_k\|) + \gamma_2^\alpha + \gamma_3^\alpha\|x_k - x^*\|, \end{aligned} \tag{27}$$

where for the third inequality we combined (9d) with Jensen's inequality (which, for a scalar $a \geq 0$, gives $E[\sqrt{a}] \leq \sqrt{E[a]}$) and the last inequality we used $\|\nabla f(x)\| = \|\nabla f(x) - \nabla f(x^*)\| \leq L_1\|x - x^*\|$ and we defined $\gamma_2^\alpha := \frac{1}{2}\alpha\delta L_1(n+6)^{3/2}$ and $\gamma_3^\alpha := \sqrt{2(n+4)}\alpha L_1$.

C. Bound for $E[\|\sigma_{k+1} - \mathcal{J}\sigma_{k+1}\|]$

We have the estimate:

$$\begin{aligned} E_{u_k}[\|\sigma_{k+1} - \mathcal{J}\sigma_{k+1}\|] &\overset{(12)}{\leq} \|\mathcal{A}\sigma_k - \mathcal{J}\mathcal{A}\sigma_k\| + E_{u_k}[\|(I-\mathcal{J})(\phi_v(x_{k+1}) - \phi_v(x_k))\|] \\ &\overset{(4)}{\leq} \rho_A\|\sigma_k - \mathcal{J}\sigma_k\| + E_{u_k}[\|\phi_v(x_{k+1}) - \phi_v(x_k)\|] \\ &\overset{(A2b)}{\leq} \rho_A\|\sigma_k - \mathcal{J}\sigma_k\| + L_\phi E_{u_k}[\|x_{k+1} - x_k\|] \\ &\overset{(27)}{\leq} \rho_A\|\sigma_k - \mathcal{J}\sigma_k\| + L_\phi\gamma_1^\alpha\|p_k - \mathcal{J}p_k\| + L_\phi\gamma_1^\alpha\|z_k - \mathcal{J}z_k\| + L_\phi\gamma_2^\alpha + L_\phi\gamma_3^\alpha\|x_k - x^*\|, \end{aligned} \tag{28}$$

where for the second inequality we used $\|I - \mathcal{J}\| = 1$.

D. Bound for $E[\|s_{k+1} - \mathcal{J}s_{k+1}\|]$

Notice that s_{k+1} has two sources of stochasticity at time k: u_k and u_{k+1}. We thus have:

$$\begin{aligned} E_{u_k,u_{k+1}}[\|s_{k+1} - \mathcal{J}s_{k+1}\|] &\overset{(12)}{\leq} \|\mathcal{A}s_k - \mathcal{J}\mathcal{A}s_k\| + E_{u_k,u_{k+1}}[\|(I-\mathcal{J})(\phi_v(x_{k+1} + \delta u_{k+1}) - \phi_v(x_k + \delta u_k))\|] \\ &\overset{(4)}{\leq} \rho_A\|s_k - \mathcal{J}s_k\| + E_{u_k,u_{k+1}}[\|\phi_v(x_{k+1} + \delta u_{k+1}) - \phi_v(x_k + \delta u_k)\|] \\ &\overset{(A2b)}{\leq} \rho_A\|s_k - \mathcal{J}s_k\| + L_\phi E_{u_k}[\|x_{k+1} - x_k\|] + L_\phi\delta E_{u_k,u_{k+1}}[\|u_{k+1} - u_k\|] \\ &\overset{(27)}{\leq} \rho_A\|s_k - \mathcal{J}s_k\| + L_\phi\gamma_1^\alpha\|p_k - \mathcal{J}p_k\| + L_\phi\gamma_1^\alpha\|z_k - \mathcal{J}z_k\| + L_\phi\gamma_2^\alpha + L_\phi\gamma_3^\alpha\|x_k - x^*\| + L_\phi\delta E_{u_k,u_{k+1}}[\|u_{k+1} - u_k\|], \end{aligned} \tag{29}$$

where for the second inequality we used $\|I - \mathcal{J}\| = 1$.

E. Bound for $E[\|z_{k+1} - \mathcal{J} z_{k+1}\|]$

Before bounding the desired term, notice that:

$$\begin{aligned} &E_{u_k}[\|\sigma_{k+1} - \sigma_k\|] \\ &\overset{(12)}{\leq} \|\mathcal{A}\sigma_k - \sigma_k\| + E_{u_k}[\|\phi_{\mathrm{v}}(x_{k+1}) - \phi_{\mathrm{v}}(x_k))\|] \\ &\overset{(A2b)}{\leq} \|(\mathcal{A} - I)(\sigma_k - \mathcal{J}\sigma_k)\| + L_\phi E_{u_k}[\|x_{k+1} - x_k\|] \\ &\overset{(27)}{\leq} \|A - I\|\|\sigma_k - \mathcal{J}\sigma_k\| + L_\phi \gamma_3^\alpha \|x_k - x^*\| \\ &\quad + L_\phi \gamma_1^\alpha (\|p_k - \mathcal{J} p_k\| + \|z_k - \mathcal{J} z_k\|) + L_\phi \gamma_2^\alpha, \end{aligned} \tag{30}$$

where, for the second inequality, we used the identity $\mathcal{A}\sigma_k - \sigma_k \overset{(4)}{=} (\mathcal{A} - I)(\sigma_k - \mathcal{J}\sigma_k)$. We then have:

$$\begin{aligned} &E_{u_k}[\|z_{k+1} - \mathcal{J} z_{k+1}\|] \\ &\overset{(12)}{\leq} \|\mathcal{A} z_k - \mathcal{J}\mathcal{A} z_k\| \\ &\quad + E_{u_k}[\|(I - \mathcal{J})(\tilde{f}_{\mathrm{v}}(x_{k+1}, \sigma_{k+1}) - \tilde{f}_{\mathrm{v}}(x_k, \sigma_k))\|] \\ &\overset{(4),(A2c)}{\leq} \rho_A \|z_k - \mathcal{J} z_k\| \\ &\quad + \tilde{L}_0 (E_{u_k}[\|x_{k+1} - x_k\|] + E_{u_k}[\|\sigma_{k+1} - \sigma_k\|]) \\ &\overset{(27),(30)}{\leq} \gamma_4^\alpha \|x_k - x^*\| + \gamma_5 \|\sigma_k - \mathcal{J}\sigma_k\| \\ &\quad + (\rho_A + \gamma_6^\alpha)\|z_k - \mathcal{J} z_k\| + \gamma_6^\alpha \|p_k - \mathcal{J} p_k\| + \gamma_7^\alpha, \end{aligned} \tag{31}$$

where in the last step we defined $\gamma_4^\alpha := \tilde{L}_0 \gamma_3^\alpha (1 + L_\phi)$, $\gamma_5 := \tilde{L}_0 \|A - I\|$, $\gamma_6^\alpha := \tilde{L}_0 \gamma_1^\alpha (1 + L_\phi)$, and $\gamma_7^\alpha := \tilde{L}_0 \gamma_2^\alpha (1 + L_\phi)$.

F. Bound for $E[\|p_{k+1} - \mathcal{J} p_{k+1}\|]$

We first need the following intermediate bound:

$$\begin{aligned} &E_{u_k, u_{k+1}}[\|s_{k+1} - s_k\|] \\ &\overset{(12)}{\leq} \|\mathcal{A} s_k - s_k\| \\ &\quad + E_{u_k}[\|\phi_{\mathrm{v}}(x_{k+1} + \delta u_{k+1}) - \phi_{\mathrm{v}}(x_k + \delta u_k))\|] \\ &\overset{(A2b)}{\leq} \|(\mathcal{A} - I)(s_k - \mathcal{J} s_k)\| \\ &\quad + L_\phi E_{u_k}[\|x_{k+1} - x_k\|] + L_\phi \delta E_{u_k, u_{k+1}}[\|u_{k+1} - u_k\|] \\ &\overset{(27)}{\leq} \|A - I\|\|s_k - \mathcal{J} s_k\| + L_\phi \gamma_3^\alpha \|x_k - x^*\| \\ &\quad + L_\phi \gamma_1^\alpha (\|p_k - \mathcal{J} p_k\| + \|z_k - \mathcal{J} z_k\|) + L_\phi \gamma_2^\alpha, \\ &\quad + L_\phi \delta E_{u_k, u_{k+1}}[\|u_{k+1} - u_k\|]. \end{aligned} \tag{32}$$

Then,

$$\begin{aligned} &E_{u_k, u_{k+1}}[\|p_{k+1} - \mathcal{J} p_{k+1}\|] \\ &\overset{(12)}{\leq} \|\mathcal{A} p_k - \mathcal{J}\mathcal{A} p_k\| \\ &\quad + E_{u_k, u_{k+1}}[\|(I - \mathcal{J})(\tilde{f}_{\mathrm{v}}(x_{k+1} + \delta u_{k+1}, s_{k+1}) \\ &\qquad\qquad - \tilde{f}_{\mathrm{v}}(x_k + \delta u_k, s_k))\|] \\ &\overset{(4),(A2c)}{\leq} \rho_A \|p_k - \mathcal{J} p_k\| + 2\tilde{L}_0 E_{u_k}[\|x_{k+1} - x_k\|] \\ &\quad + \tilde{L}_0 E_{u_k, u_{k+1}}[\|s_{k+1} - s_k\|] \\ &\quad + \tilde{L}_0 \delta E_{u_k, u_{k+1}}[\|u_{k+1} - u_k\|] \\ &\overset{(27),(30)}{\leq} \gamma_4^\alpha \|x_k - x^*\| + \gamma_5 \|s_k - \mathcal{J} s_k\| \\ &\quad + \gamma_6^\alpha \|z_k - \mathcal{J} z_k\| + (\rho_A + \gamma_6^\alpha)\|p_k - \mathcal{J} p_k\| \\ &\quad + \gamma_7^\alpha + \gamma_8 E_{u_k, u_{k+1}}[\|u_{k+1} - u_k\|], \end{aligned} \tag{33}$$

where we defined $\gamma_8 := \tilde{L}_0 \delta (1 + L_\phi)$.

G. Proof of Theorem 7

By combining the estimates (26), (28), (29), (31), and (33), we conclude that (23) holds with

$$M(\alpha) = \begin{bmatrix} \sqrt{1 - \beta_1^\alpha} & 0 & 0 & \gamma_1^\alpha & \gamma_1^\alpha \\ L_\phi \gamma_3^\alpha & \rho_A & 0 & L_\phi \gamma_1^\alpha & L_\phi \gamma_1^\alpha \\ L_\phi \gamma_3^\alpha & 0 & \rho_A & L_\phi \gamma_1^\alpha & L_\phi \gamma_1^\alpha \\ \gamma_4^\alpha & \gamma_5 & 0 & \rho_A + \gamma_6^\alpha & \gamma_6^\alpha \\ \gamma_4^\alpha & 0 & \gamma_5 & \gamma_6^\alpha & \rho_A + \gamma_6^\alpha \end{bmatrix} \tag{34}$$

and

$$b = \begin{bmatrix} \beta_2^\alpha \\ L_\phi \gamma_2^\alpha \\ L_\phi \gamma_2^\alpha + L_\phi \delta E_{u_k, u_{k+1}}[\|u_{k+1} - u_k\|] \\ \gamma_7^\alpha \\ \gamma_7^\alpha + \gamma_8 E_{u_k, u_{k+1}}[\|u_{k+1} - u_k\|] \end{bmatrix}.$$

Notice that b satisfies $\|b\| \leq \mathcal{O}(\delta \cdot E[\|u_{k+1} - u_k\|^2])$. By substituting the expressions of the constants involved, we have that, for[5] $\alpha \to 0^+$,

$$M(\alpha) = \begin{bmatrix} \sqrt{1 - \alpha\mu} & 0 & 0 & 0 & 0 \\ 0 & \rho_A & 0 & 0 & 0 \\ 0 & 0 & \rho_A & 0 & 0 \\ 0 & \gamma_5 & 0 & \rho_A & 0 \\ 0 & 0 & \gamma_5 & 0 & \rho_A \end{bmatrix} + o(\alpha),$$

which proves that the spectral radius of $M(\alpha)$ is smaller than 1 for sufficiently small $\alpha > 0$. To determine an estimate on the largest value of α that guarantees $\rho(M(\alpha)) < 1$, we leverage the Gershgorin Circle Theorem. An application of the theorem to row 1 of $M(\alpha)$ gives the condition:

$$\sqrt{1 - \alpha\mu\,(1 - 2\alpha(n+4)L_1)} < 1 - \frac{2\alpha\sqrt{n}}{\delta}. \tag{35a}$$

Squaring both sides of (35a) and simplifying yields:

$$\mu(1 - 2\alpha(n+4)L_1) > \frac{4\sqrt{n}}{\delta} - \frac{4\alpha n}{\delta^2}.$$

The condition $\alpha < \frac{1}{2(n+4)L_1}$ ensures that the left-hand side is positive, while the assumption $\delta \leq \alpha\sqrt{n}$ guarantees that the right-hand side is non-negative. Hence, under the assumptions of Theorem 7, the above inequality holds, and consequently (35a) is also satisfied. Applying Gershgorin's theorem to rows 2–3 of $M(\alpha)$ gives:

$$\rho_A + \alpha L_\phi \left(\sqrt{2(n+4)} L_1 + \frac{2\sqrt{n}}{\delta} \right) < 1 \tag{35b}$$

which gives the estimate α_1^*; applying Gershgorin's theorem to rows 4-5 of $M(\alpha)$ gives:

$$\begin{aligned} &\rho_A + \tilde{L}_0 \|A - I\| \\ &\quad + \tilde{L}_0 (1 + L_\phi)\, \alpha \left(\frac{2\sqrt{n}}{\delta} + \sqrt{2(n+4)} L_1 \right) < 1 \end{aligned} \tag{35c}$$

which gives the estimate α_2^*.

[5] We say $f(x) = o(g(x))$ as $x \to a$ if $\lim_{x \to a} \frac{f(x)}{g(x)} = 0$.

We have thus shown that the estimate (19) holds, where $M(\alpha)$ is a Schur-stable matrix with spectral radius $\eta = \rho(M(\alpha))$, and b is a vector satisfying $\|b\| = \mathcal{O}(\delta\,\mathbb{E}[\|u_{k+1} - u_k\|^2])$. Hence, the bound (19) is established with $\varepsilon = \mathcal{O}(\delta\,\mathbb{E}[\|u_{k+1} - u_k\|^2])$, consistent with the first identity in (20).

Finally, we are left to prove the last identity in (20); that is, $\|b\| = \mathcal{O}(\delta n)$. To this end, we apply the inequality $(a+b)^2 \le 2a^2 + 2b^2$ and bound the norm of the vector b as:

$$\begin{aligned}\|b\| &\le \delta\Bigg[\alpha L_1 n + \frac{\alpha^2 L_1^2}{2}(n+6)^3 \\ &\quad + 2\left(L_\phi^2 + \tilde{L}_0^2\,(1+L_\phi)^2\right) E[\|u_{k+1}-u_k\|^2] \\ &\quad + \frac{3}{4}\alpha^2 L_1^2 (n+6)^3 \left(L_\phi^2 + \tilde{L}_0^2\,(1+L_\phi)^2\right)\Bigg]^{1/2} \\ &\le \delta\Bigg[\alpha L_1 n + 8n^2\left(L_\phi^2 + \tilde{L}_0^2\,(1+L_\phi)^2\right) \\ &\quad + \alpha^2 L_1^2 (n+6)^3 \left(\frac{1}{2} + \frac{3}{4}L_\phi^2 + \frac{3}{4}\tilde{L}_0^2\,(1+L_\phi)^2\right)\Bigg]^{1/2} \\ &= \delta\mathcal{O}(\max\{\sqrt{\alpha n}, n, \alpha n^{3/2}\}).\end{aligned} \tag{36}$$

where the last inequality follows from $E[\|u_{k+1} - u_k\|^2] \le 2n$ (which follows from (7), since u_{k+1} and u_k are independent). By noting that $\alpha = \mathcal{O}(\frac{1}{n})$, it follows that the second term dominates the maximization for large n and the estimate $\|b\| \le \varepsilon = \mathcal{O}(\delta n)$ follows.

H. Proof of Theorem 8

The estimate (21) follows by noting that, from (23), η can be selected to be an upper bound for the spectral radius of $M(\alpha)$ and, by using (34) and (35), (21) follows. Finally, the expression (22) follows from (23) and the estimate (24).

V. Numerical and experimental validation on robotic formation control problems

In this section, we demonstrate the applicability of the framework through numerical simulations and experiments. We consider a multi-agent robotic formation control problem (see Fig. 2(Left)), where each agent i represents a robot with planar position $x_i \in \mathbb{R}^2$, aiming to reach its privately known *target* $r_i \in \mathbb{R}^2$. Moreover, the robots seek to maintain swarm cohesion, which is enforced by penalizing large deviations of x_i from the swarm *barycenter* $\sigma_{\mathrm{f}}(x) := \frac{1}{N}\sum_{i=1}^N x_i$. These objectives can be modeled using an instance of (1) with $\phi_i(x_i) = x_i$ and

$$\tilde{f}_i(x_i, \sigma_{\mathrm{f}}(x)) = \frac{\gamma_i}{2}\|x_i - r_i\|^2 + \frac{1}{2}\|x_i - \sigma_{\mathrm{f}}(x)\|^2,$$

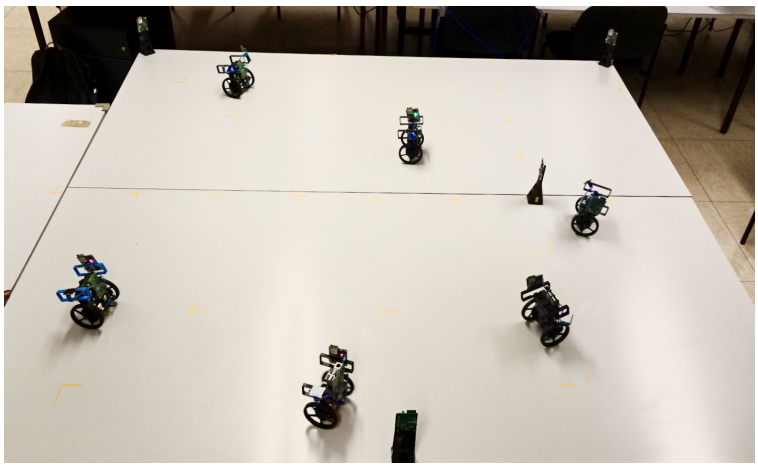
where $\gamma_i > 0$ models the extent to which robot i prioritizes reaching its target, over maintaining cohesion with the swarm. In the remainder, we use $N = 5$ and $\gamma_i = 2\ \forall i$.

A. Numerical simulations

For the numerical simulations, both the target positions r_i and the initial positions x_i^0 were generated uniformly at random within the interval $[0, 10]$. The communication topology was chosen as a random graph with an Erdős–Rényi topology, using an edge probability $p = 0.6$ and uniform edge weights, normalized to satisfy Assumption 1. Simulation results from an implementation of Algorithm 1 are presented in Fig.s 1 and 2. The algorithm's parameters have been chosen as follows: $\alpha = 2 \cdot 10^{-3}$, $\delta = 10^{-5}$. The top plot of Fig. 1 shows that `ARGFree` (orange line) enables the agents to reach the desired configuration with a relative function value error on the order of 10^{-1}. The figure also proposes a comparison with the method from [8] (green line), which relies on exact gradient information. Compared to our approach, exact gradient methods exhibit faster convergence and higher accuracy, though they cannot be implemented under requirements (R1)–(R2). The plot further includes a modified version of `ARGFree` (blue line), in which the exploration signal is filtered through a damping term of the form $u_i^{k+1} = B_i u_i^k + v_i^{k+1}$, where $B_i \in \mathbb{R}^{2\times 2}$ satisfies $0.9 < \rho(B_i) < 1$, and $v_i \sim \mathcal{N}(0, 0.16 I_2)$. Intuitively, this damping mitigates abrupt variations in the exploration signal u_i^k, allowing the tracking variables s_i^k and p_i^k to better follow the perturbed function evaluations they are designed to track. This yields a reduction in the relative function value error, as illustrated in Fig. 1(Top).

The bottom plot of Fig. 1 illustrates the methods' behavior in the presence of measurement noise. Specifically, noise was introduced by replacing each x_i with $w_i x_i$ in the right-hand side of lines 1–7 of Algorithm 1, where $w_i \sim \mathcal{N}(0, 0.2 I_2)$, modeling multiplicative noise in the localization sensors. The results in Fig. 1(Bottom) show that, in the presence of noise, `ARGFree` achieves a comparable convergence rate and superior steady-state accuracy relative to the exact gradient-based method. This highlights one of the main advantages of the proposed approach: it requires not only less information, but also outperforms exact gradient techniques under noisy conditions. This behavior can be attributed to the fact that, in exact gradient-based methods, errors in the positions x_i tend to be amplified through gradient evaluations, whereas `ARGFree` relies directly on loss evaluations, thus operating directly on the performance metric to be minimized and implicitly neglecting the effect of noise on the gradients.

Fig. 2 illustrates the paths followed by the robots. Specifically, it compares the exact gradient-based method from [8] (center) with the modified version of `ARGFree` (right). Both algorithms drive the robots to asymptotic configurations (marked by '$\times$' symbols) that balance reaching their individual targets (denoted by '$\star$' symbols) and maintaining proximity to the swarm barycenter (denoted by the '$+$' symbol). As expected, the trajectories generated by `ARGFree` are more irregular due to the use of randomized perturbations for gradient estimation; however, the robots converge asymptotically to (visually) nearly identical configurations. This demonstrates that the proposed approach achieves com-



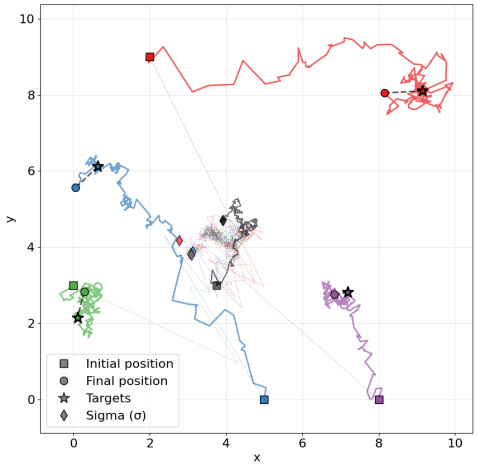


Fig. 3. Results from the experimental implementation of ARGFree on a set of agents equipped with Raspberry Pi embedded systems. **(Left)** Photo of the experimental setup. Each robot is a Balboa 32U4 platform equipped with a Raspberry Pi Zero 2 W module and a Decawave DWM1001 module for localization. Localization is achieved via the Decawave modules, which estimate relative distances to four fixed anchors (at known positions) using range measurements based on Time-of-Arrival (ToA) information. **(Right)** Paths followed by the robots during the experiment. See Section V-B for details and discussion.

parable performance to exact gradient methods despite not requiring gradient information for its implementation.

B. Experimental results

We now present results from an experimental implementation of ARGFree on a set of agents equipped with embedded communication an localization systems[6]. The experimental setup, illustrated in Fig. 3, consists of Balboa 32U4 robots, each equipped with a Raspberry Pi Zero 2 W module and a Decawave DWM1001 ultra-wideband (UWB) module. Inter-agent communication is implemented via the Raspberry Pi's Bluetooth 5.0 interface using *Bluetooth Classic*, while localization is performed by the Decawave modules, which estimate relative distances to fixed anchors (placed at known positions) based on range measurements. Specifically, distances are derived from Time-of-Arrival (ToA) data, and the Decawave firmware performs multilateration using Time-Difference-of-Arrival (TDoA) information to locally estimate each agent's position.

Simulation results of the ARGFree implementation with $\alpha = 2\times10^{-2}$ and $\delta = 10^{-3}$ on the described setup are shown in Fig. 3(right). Experiments were conducted over a duration of 2 minutes and 10 seconds, corresponding to 500 iterations, and featured asynchronous updates with delayed position readings from each robot. To focus on the algorithmic performance of ARGFree without additional complexities arising from robot motion control, the robots' movements were simulated. The results in Fig. 3(Right) indicate that each robot converges to a position representing a trade-off between its desired target r_i and the evolving barycenter $\sigma(x)$. Despite the presence of noise due to randomized perturbations, ARGFree exhibited satisfactory convergence and robustness in this experimental setup, showcasing the potential of the method in this application.

[6]The code used in the experiments is available at https://github.com/speciale7/gf_dist_opt.

VI. Conclusions

We have proposed ARGFree, a distributed algorithm for solving aggregative cooperative optimization problems without requiring explicit gradient information. The method relies on randomized finite-difference approximations of the cost gradient. We established that the algorithm converges to a neighborhood of the optimizer, whose size can be controlled through an appropriate choice of the smoothing parameter. Experimental validation on a robotic formation control problem, conducted on a team of embedded systems built on Segway-type robots, demonstrated the effectiveness of the proposed method. This work opens the opportunity for several directions of future work, including the use of single-point and multi-point gradient approximations, and adaptations of the methods in feedback optimization configurations.

Appendix I
Proof of Lemma 5

We begin by proving $\bar{\sigma}_k = \sigma_f(x_k)$. By multiplying by $\frac{1}{N}\mathbb{1}^\top$ both sides of (12b) and by using $\frac{1}{N}\mathbb{1}^\top A = \frac{1}{N}\mathbb{1}^\top$ (from Assumption 1), we have:

$$\bar{\sigma}_{k+1} = \bar{\sigma}_k + \frac{1}{N}\sum_{i=1}^{N}\left(\phi_i(x_i^{k+1}) - \phi_i(x_i^k)\right).$$

By telescoping the sum:

$$\begin{aligned}\bar{\sigma}_{k+1} &= \bar{\sigma}_0 + \frac{1}{N}\sum_{i=1}^{N}\left(\phi_i(x_i^{k+1}) - \phi_i(x_i^0)\right)\\ &= \frac{1}{N}\sum_{i=1}^{N}\phi_i(x_i^{k+1}),\end{aligned}$$

where the last identity follows from the choice of initial conditions $\sigma_i^0 = \phi_i(x_i^0)$. The first assertion thus follows

by definition of $\sigma_{\rm f}(x)$ (see (1)). The proof of the remaining assertions follows by iterating the argument.

APPENDIX II
PROOF OF LEMMA 6

We begin by recalling the following basic properties: (a) $f(x) - f(x^\star) \leq \nabla f(x)^T (x - x^\star)$, which follows from convexity of $f(x)$; and (b) $\|\nabla f(x)\|^2 \leq 2L_1 \left(f(x) - f(x^\star)\right)$ which follows from Lipschitz smoothness. We have:

$$
\begin{aligned}
&E_u[\|x - \alpha g_\delta(x) - x^*\|^2] \\
&\quad = \|x - x^*\|^2 - 2\alpha \left\langle E_u[g_\delta(x)], x - x^* \right\rangle + \alpha^2 E_u[\|g_\delta(x)\|^2] \\
&\quad \overset{\text{(9d)}}{\leq} \|x - x^*\|^2 - 2\alpha \left\langle E_u[g_\delta(x)], x - x^* \right\rangle \\
&\qquad + \alpha^2 \left[\frac{\delta^2 (n+6)^3}{2} L_1^2 + 2(n+4) \|\nabla f(x)\|^2 \right] \\
&\quad \overset{\text{(8),(a)}}{\leq} \|x - x^*\|^2 - 2\alpha \left(f_\delta(x) - f_\delta(x^\star) \right) \\
&\qquad + \alpha^2 \left[\frac{\delta^2 (n+6)^3}{2} L_1^2 + 2(n+4) \|\nabla f(x)\|^2 \right] \\
&\quad \overset{\text{(9b),(b)}}{\leq} \|x - x^*\|^2 - 2\alpha (f(x) - f_\delta(x^*)) \\
&\qquad + \alpha^2 \left[\frac{\delta^2 (n+6)^3}{2} L_1^2 + 4(n+4) L_1 (f(x) - f(x^*)) \right] \\
&\quad \overset{\text{(9a)}}{\leq} \|x - x^*\|^2 - 2\alpha \left(1 - 2\alpha(n+4)L_1\right) \left(f(x) - f(x^*)\right) \\
&\qquad + \delta^2 n \alpha L_1 + \frac{\delta^2 (n+6)^3}{2} \alpha^2 L_1^2 \\
&\quad \leq \|x - x^*\|^2 - \alpha\mu (1 - 2\alpha(n+4)L_1) \|x - x^*\|^2 \\
&\qquad + \delta^2 n \alpha L_1 + \frac{\delta^2 (n+6)^3}{2} \alpha^2 L_1^2,
\end{aligned}
$$

where for the last inequality we used $f(x) - f(x^*) \geq \mu/2 \|x - x^*\|^2$. This establishes that

$$
E_u[\|x - \alpha g_\delta(x) - x^*\|^2] \leq (1 - \beta_1^\alpha) \|x - x^*\|^2 + (\beta_2^\alpha)^2. \tag{37}
$$

To conclude, notice that

$$
\begin{aligned}
E_u[\|x - \alpha g_\delta(x) - x^*\|] &\leq \left(E_u[\|x - \alpha g_\delta(x) - x^*\|^2] \right)^{1/2} \\
&\overset{(37)}{\leq} \sqrt{1 - \beta_1^\alpha} \|x - x^*\| + \beta_2^\alpha,
\end{aligned}
$$

where the first bound follows by Jensen's inequality, and the second one from $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a, b \geq 0$.

REFERENCES

[1] A. Nedić and J. Liu, "Distributed optimization for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 77–103, 2018. doi: 10.1146/annurev-control-060117-105131

[2] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019. doi: 10.1016/j.arcontrol.2019.05.006

[3] G. Notarstefano, I. Notarnicola, and A. Camisa, "Distributed optimization for smart cyber-physical networks," *Foundations and Trends in Systems and Control*, vol. 7, no. 3, pp. 253–383, 2019. doi: 10.1561/9781680836196

[4] A. Testa, G. Carnevale, and G. Notarstefano, "A tutorial on distributed optimization for cooperative robotics: from setups and algorithms to toolboxes and research directions," *Proceedings of the IEEE*, 2025. doi: 10.1109/jproc.2025.3557698/mm1

[5] A. Mehrnoosh and G. Bianchin, "Optimization of linear multi-agent dynamical systems via feedback distributed gradient descent methods," *arXiv preprint*, Jul. 2025, arXiv:2403.18386.

[6] G. Carnevale, N. Mimmo, and G. Notarstefano, "Nonconvex distributed feedback optimization for aggregative cooperative robotics," *Automatica*, vol. 167, p. 111767, 2024. doi: 10.1016/j.automatica.2024.111767

[7] G. Bianchin, M. Vaquero, J. Cortés, and E. Dall'Anese, "k-dimensional agreement in multiagent systems," *IEEE Transactions on Automatic Control*, vol. 69, no. 12, pp. 8978–08 985, Dec. 2024. doi: 10.1109/TAC.2024.3431108

[8] X. Li, L. Xie, and Y. Hong, "Distributed aggregative optimization over multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 67, no. 6, pp. 3165–3171, 2022. doi: 10.1109/tac.2021.3095456

[9] S. Liang, P. Yi, and Y. Hong, "Distributed Nash equilibrium seeking for aggregative games with coupled constraints," *Automatica*, vol. 85, pp. 179–185, 2017. doi: 10.1016/j.automatica.2017.07.064

[10] X. Li, X. Yi, and L. Xie, "Distributed online convex optimization with an aggregative variable," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 1, pp. 438–449, 2021. doi: 10.1109/tcns.2021.3107480

[11] T. Wang and P. Yi, "Distributed projection-free algorithm for constrained aggregative optimization," *International Journal of Robust and Nonlinear Control*, vol. 33, no. 10, pp. 5273–5288, 2023. doi: 10.1002/rnc.6640

[12] L. Chen, G. Wen, X. Fang, J. Zhou, and J. Cao, "Achieving linear convergence in distributed aggregative optimization over directed graphs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 7, pp. 4529–4541, 2024. doi: 10.1109/tsmc.2024.3382173

[13] G. Carnevale and G. Notarstefano, "A learning-based distributed algorithm for personalized aggregative optimization," in *IEEE Conf. on Decision and Control*, 2022, pp. 1576–1581. doi: cdc51059.2022.9992678

[14] R. Brumali, G. Carnevale, and G. Notarstefano, "Data-driven distributed optimization via aggregative tracking and deep-learning," *arXiv preprint*, 2025, arXiv:2503.04668.

[15] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, vol. 17, pp. 527–566, 2017. doi: 10.1007/s10208-015-9296-2

[16] Y. Tang, J. Zhang, and N. Li, "Distributed zero-order algorithms for nonconvex multiagent optimization," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 1, pp. 269–281, 2020. doi: 10.1109/tcns.2020.3024321

[17] D. Hajinezhad, M. Hong, and A. Garcia, "ZONE: Zeroth-order nonconvex multiagent optimization over networks," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 3995–4010, 2019. doi: 10.1109/TAC.2019.2896025

[18] E. Mhanna and M. Assaad, "Single point-based distributed zeroth-order optimization with a non-convex stochastic objective function," in *International Conference on Machine Learning*. PMLR, 2023, pp. 24 701–24 719.

[19] J. Lu and C. Y. Tang, "Zero-gradient-sum algorithms for distributed convex optimization: The continuous-time case," *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2348–2354, 2012. doi: 10.1109/acc.2011.5991466

[20] Y. Pang and G. Hu, "Randomized gradient-free distributed optimization methods for a multiagent system with unknown cost function," *IEEE Transactions on Automatic Control*, vol. 65, no. 1, pp. 333–340, 2019. doi: 10.1109/tac.2019.2914025

[21] D. Yuan and D. W. Ho, "Randomized gradient-free method for multiagent optimization over time-varying networks," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 6, pp. 1342–1347, 2014. doi: 10.1109/tnnls.2014.2336806

[22] A. K. Sahu and S. Kar, "Decentralized zeroth-order constrained stochastic optimization algorithms: Frank–wolfe and variants with applications to black-box adversarial attacks," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 1890–1905, 2020. doi: 10.1109/jproc.2020.3012609

[23] D. Wang, J. Zhou, Z. Wang, and W. Wang, "Random gradient-free optimization for multiagent systems with communication noises under a time-varying weight balanced digraph," *IEEE Transactions on*

Systems, Man, and Cybernetics: Systems, vol. 50, no. 1, pp. 281–289, 2017. doi: 10.1109/tsmc.2017.2757265

[24] O. Bilenne, P. Mertikopoulos, and E. V. Belmega, "Fast optimization with zeroth-order feedback in distributed, multi-user MIMO systems," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6085–6100, 2020. doi: 10.1109/tsp.2020.3029983

[25] N. Mimmo, G. Carnevale, A. Testa, and G. Notarstefano, "Extremum seeking tracking for derivative-free distributed optimization," *IEEE Transactions on Control of Network Systems*, 2024. doi: 10.1109/TCNS.2024.3510368 (Early access).

[26] Y. Huang and J. Hu, "Zeroth-order learning in continuous games via residual pseudogradient estimates," *IEEE Transactions on Automatic Control*, 2024. doi: 10.1109/tac.2024.3479874 (Early access).

[27] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *IEEE Conf. on Decision and Control*, 2005, pp. 2996–3000. doi: 10.1109/cdc.2005.1582620

[28] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004. doi: 10.23919/ecc.2007.7068297

[29] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *American Control Conference*, Portland, OR, Jun. 2005, pp. 1859–1864. doi: 10.1109/acc.2005.1470239

[30] B. Gharesifard and J. Cortés, "Distributed strategies for generating weight-balanced and doubly stochastic digraphs," *European Journal of Control*, vol. 18, no. 6, pp. 539–557, 2012. doi: 10.3166/ejc.18.539-557

[31] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985. ISBN 0521386322