Direct-search methods for decentralized blackbox optimization

E. Bergou[*] Y. Diouane [†] V. Kungurtsev [‡] C. W. Royer [§]

April 5, 2025

Abstract

Derivative-free optimization algorithms are particularly useful for tackling blackbox optimization problems where the objective function arises from complex and expensive procedures that preclude the use of classical gradient-based methods. In contemporary decentralized environments, such functions are defined locally on different computational nodes due to technical or privacy constraints, introducing additional challenges within the optimization process.

In this paper, we adapt direct-search methods, a classical technique in derivative-free optimization, to the decentralized setting. In contrast with zeroth-order algorithms, our algorithms rely on positive spanning sets to define suitable search directions, while still possessing global convergence guarantees thanks to carefully chosen stepsizes. Numerical experiments highlight the advantages of direct-search techniques over gradient-approximation-based strategies.

Keywords: Decentralized optimization; derivative-free optimization; decentralized direct-search; global convergence.

1 Introduction

Decentralized optimization (also referred to as distributed, network, or consensus optimization in the literature) has been an increasingly popular topic of investigation in recent years [20, Chapter 11], as networked control and learning systems have proliferated. In a decentralized setting, an objective function is defined as a sum of functions wherein each individual function is known only locally to some agent, and the problem can be solved only through peer-to-peer communication. Note that this is distinct from another form of distributed, namely "federated", optimization, wherein a central node uses subsidiary worker notes to ease the computational burden, but otherwise coordinates the procedure [7]. Decentralized optimization algorithms work

[*] Mohammed VI Polytechnic University, Ben Guerir, Morocco. (elhoucine.bergou@um6p.ma).

[†] GERAD and Department of Mathematics and Industrial Engineering, Polytechnique Monteal, Montreal, Canada. (youssef.diouane@polymtl.ca). Funding for this author's research was partially provided by the NSERC Discovery grant (RGPIN-2024-0509).

[‡] Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague. (vyacheslav.kungurtsev@fel.cvut.cz).

[§] LAMSADE, CNRS, Université Paris Dauphine-PSL, 75016 Paris, France. (clement.royer@lamsade.dauphine.fr). Funding for this author's research was partially provided by CNRS under the IEA grant BONUS and by Agence Nationale de la Recherche through program ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

at the agent level by updating local copies of the problem variables individually, then combining these updates with information obtained through communications. The most classical algorithm of that form is decentralized gradient descent (DGD), for which a number of convergence results have been derived in the convex setting [19, 27]. The nonconvex case proved significantly more challenging, leading to the development of another class of algorithms termed gradient tracking [4, 17, 23]. Still, convergence guarantees have also been derived for DGD techniques in the nonconvex setting, either by relying on carefully chosen step sizes [28] or focusing on subclasses of communication networks [24]. A key distinction between the aforementioned results and their centralized counterparts lies in the fact that each agent possesses its own copy of the problem variables. As a result, a globally convergent method should guarantee that all copies are eventually in agreement, i.e. that *consensus* is reached among all agents. This property is typically obtained through the analysis of an appropriate Lyapunov function, possibly used within the algorithm itself [20, Chapter 11]. An alternate approach consists in ensuring approximate consensus by a penalty reformulation of the problem, where the level of consensus is controlled by the penalty parameter [28]. In both cases, improvement of the objective function is obtained for each agent by taking steps in the negative gradient directions.

When derivatives of the local function are unavailable, the decentralized optimization literature has focused on using zeroth-order algorithms, that estimate gradients or directional derivatives through finite-difference type estimates [3, 8, 11, 16, 21, 22, 25]. Although a common choice to alleviate the absence of explicit gradient values, zeroth-order approaches are only a subset of derivative-free optimization, a field that has grown in importance due to multiple applications in engineering simulations and parameter tuning [1, 2, 15]. Among derivative-free optimization techniques, direct-search algorithms proceed by exploring the variable space through suitably chosen directions. Even though those directions are chosen without gradient knowledge, the resulting algorithms can be endowed with a rich convergence analysis [6, 14]. In addition, direct-search schemes have been successfully implemented in parallel environments, with convergence guarantees being established in both synchronous and asynchronous settings [10, 12, 13]. However, to the best of our knowledge, these results only consider a centralized setting, and an investigation (both theoretical and practical) of direct-search methods on decentralized optimization problems has yet to be conducted.

In this paper, we propose variants on the direct-search paradigm dedicated to solving decentralized optimization problems. We preserve key features of direct-search schemes, such as the use of positive spanning sets to explore the variable space and sufficient decrease conditions, but we adapt stepsize conditions in order to guarantee convergence despite the decentralized environment. Our analysis either guarantees convergence to at least a stationary point of a certain penalty function corresponding to the problem, or certifies consensus in the limit. Our numerical experiments, in which we adapt a standard DFO benchmark to the decentralized setting, show promising performance of direct-search schemes compared to zeroth-order strategies.

The rest of this paper is organized as follows. Section 2 formalizes the decentralized optimization setting, and provides background material on decentralized gradient descent and its zeroth-order variants. Section 3 details two direct-search proposals based on adapting the decentralized gradient iteration. Section 4 contains global convergence results for the two proposed methods. Section 5 reports numerical comparisons between our algorithms and zeroth-order strategies. We discuss our findings in Section 6.

Notations In the rest of the paper, $\|x\|$ will denote the Euclidean norm of the vector x; we will use the same notation for the induced operator norm on matrices. For any $r \in \mathbb{N}$, the vector of ones in $\mathbb{R}^r$ will be indicated by $\mathbf{1}_r$, while $\mathbf{I}_r$ will denote the identity matrix in $\mathbb{R}^{r \times r}$. Given two matrices $A \in \mathbb{R}^{n_1 \times n_2}$ and $B \in \mathbb{R}^{n_3 \times n_4}$, $C := A \otimes B$ denotes the Kronecker product of A and B, i.e. the matrix $C \in \mathbb{R}^{n_1 n_3 \times n_2 n_4}$ given by $C = \begin{bmatrix} a_{11}B & \cdots & a_{1n_2}B \\ \vdots & \ddots & \vdots \\ a_{n_1 1}B & \cdots & a_{n_1 n_2}B \end{bmatrix}$.

2 Background on decentralized optimization

In this paper, we are interested in the following optimization problem

$$\min_{x \in \mathbb{R}^n} \quad \sum_{i=1}^{m} f_i(x), \tag{1}$$

where every function f_i is smooth but its derivative is unavailable for algorithmic use. Throughout this paper, we make the following standard assumptions about the objective function.

Assumption 2.1 *For every $i \in \{1, \ldots, m\}$, the function f_i is continuously differentiable and the gradient ∇f_i is L_i-Lipschitz continuous.*

Assumption 2.2 *The function $x \mapsto \sum_{i=1}^m f_i(x)$ is bounded from below by $f_{\text{low}} \in \mathbb{R}$.*

In a decentralized setting, the problem (1) is to be solved over a network of m agents modeled by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, m\}$ represents the set of agents and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the set of edges. Each agent i can evaluate the function f_i, but is unaware of the other functions that define the objective. In order to solve problem (1) over the network, every agent $i \in \mathcal{V}$ thus maintains its own copy of the vector of parameters, denoted by $x_i \in \mathbb{R}^n$. Letting $\mathbf{x} := \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \in \mathbb{R}^{mn}$ denote the concatenation of all local copies, the original problem (1) becomes equivalent to

$$\begin{array}{ll} \min\limits_{\mathbf{x} \in \mathbb{R}^{mn}} & F(\mathbf{x}) := \sum_{i=1}^m f_i(x_i) \\ \text{subject to} & x_i = x_j \quad \forall (i,j) \in \mathcal{E}. \end{array} \tag{2}$$

To enforce the constraints in problem (2), any agent i can obtain the values of the copies of its immediate neighbours through communication. Letting $\mathcal{N}_i := \{j \in \{1, \ldots, m\} \mid j \neq i, \ (i,j) \in \mathcal{E}\}$ denote the sets of immediate neighbors of agent i, a round of communications allows agent i to receive $x_{\mathcal{N}_i} = \{x_j\}_{j \in \mathcal{N}_i}$. This information is typically combined with that of agent i by means of a *mixing matrix* $W \in \mathbb{R}^{m \times m}$ whose non-zero entries correspond to elements in $\{\mathcal{N}_i\}$, for which we enforce the following requirements.

Assumption 2.3 *The mixing matrix $W = [w_{ij}]$ is a symmetric matrix with nonnegative coefficients such that $w_{ij} > 0$ if and only if $i = j$ or $j \in \mathcal{N}_i$. It is also doubly stochastic, i.e.*

$$\sum_{i=1}^{m} w_{ij} = 1\,, \forall j \in \{1, \ldots, m\} \quad \textit{and} \quad \sum_{j=1}^{m} w_{ij} = 1\,, \forall i \in \{1, \ldots, m\}.$$

Finally, letting $\lambda_1(W) \geq \lambda_2(W) \geq \cdots \geq \lambda_n(W)$ *denote the eigenvalues of* W*, we have:*

i) $\lambda_1(W) = 1$ *and* $\lambda_2(W) < 1$*,*

ii) $W\mathbf{1} = \mathbf{1}$*,*

iii) $-1 < \lambda_n(W) \leq 0$*.*

Under Assumption 2.3, the eigenvalues of W necessary lie between -1 and 1, with the largest eigenvalue equal to 1. As a result, problem (2) is equivalent to

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^{mn}} & F(\mathbf{x}) = \sum_{i=1}^m f_i(x_i) \\ \text{subject to} & x_i = \sum_{j=1}^m w_{ij}x_j \quad \forall i = 1,\dots,m. \end{array} \tag{3}$$

It follows that any solution $\mathbf{x}^* = [x_i^*]_i \in \mathbb{R}^{nm}$ of problem (2) must satisfy $(\mathbf{I}_{nm} - \widehat{W})\mathbf{x}^* = 0$, where $\widehat{W} := W \otimes \mathbf{I}_n \in \mathbb{R}^{nm\times nm}$.

2.1 Penalty function and reformulation

A common approach to decentralized optimization consists in replacing the constrained formulation (2) by an unconstrained minimization problem of a suitable penalty function, where the penalty function is typically chosen to be quadratic and depends on the mixing matrix W [28]. The resulting optimization problem has the form

$$\min_{\mathbf{x}\in\mathbb{R}^{nm}} \quad \mathcal{L}(\mathbf{x};\gamma) := F(\mathbf{x}) + P(\mathbf{x};\gamma), \tag{4}$$

where $F(\mathbf{x}) := \sum_{i=1}^m f_i(x_i)$, $\gamma > 0$, and

$$P(\mathbf{x};\gamma) := \frac{1}{2\gamma}\|\mathbf{x}\|^2_{\mathbf{I}_{nm} - \widehat{W}} = \frac{1}{2\gamma}\mathbf{x}^{\mathrm{T}}(\mathbf{I}_{nm} - \widehat{W})\mathbf{x} = \frac{1}{2\gamma}\sum_{i=1}^m \|\widehat{x}_i - x_i\|^2,$$

with $\widehat{x}_i := \sum\limits_{j\in\mathcal{N}_i\cup\{i\}} w_{ij}x_j$. The gradient of this penalty function is given by

$$\nabla P(\mathbf{x};\gamma) = \frac{1}{\gamma}(\mathbf{I}_{nm} - \widehat{W})\mathbf{x}.$$

As $\gamma \to 0$, solutions of the penalized formulation (4) converge to that of the constrained problem (2). However, note that a stationary point of $\mathbf{x}^* = [x_i^*]$ of problem (4) is not necessarily a stationary point of problem (2), since the vectors $x_1^*,\dots,x_m^*$ need not be identical. Still, standard analyses of decentralized gradient methods establish convergence towards a stationary point of the penalty function [19, 28].

2.2 Decentralized gradient descent

Decentralized gradient descent is based on the following recursion

$$x_i^{(k+1)} = \widehat{x}_i^{(k)} - \alpha^{(k)}\nabla f_i(x_i^{(k)}) \qquad \forall k \in \mathbb{N},\ \forall i \in \{1,\dots,m\}, \tag{5}$$

where $\widehat{x}_i^{(k)} := \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} x_j^{(k)}$ and $\alpha^{(k)} > 0$ is a positive stepsize. In a decentralized setting, the stepsize sequence $\{\alpha^{(k)}\}$ is typically fixed a priori, either as a constant or a decreasing sequence, and all agents use the same sequence throughout the iterations.

A natural extension of decentralized gradient techniques in absence of derivatives consists in approximating derivatives through deterministic or randomized finite differences. The resulting algorithms, termed *zeroth-order* decentralized gradient techniques, have the form

$$x_i^{(k+1)} = \widehat{x}_i^{(k)} - \alpha^{(k)} g_i(x_i^{(k)}) \qquad \forall k \in \mathbb{N},\ \forall i \in \{1, \dots, m\}, \tag{6}$$

where $g_i(x_i^{(k)})$ is a gradient approximation. Akin to their first-order counterparts, zeroth-order decentralized gradient methods do not use function values to check for decrease in the objective. Although appropriate in a decentralized environment, this paradigm differs significantly from the dominant approach in derivative-free optimization, that consists in accepting steps that reduce the objective value, and reject those that do not [2]. In the next section, we describe two ways to adapt this approach to the decentralized setting.

3 Decentralized direct-search frameworks

In this section, we propose two ways to adapt the classical direct-search algorithmic framework [14] to solve problem (1). Section 3.1 describes a method that uses a Lyapunov function at every iteration, this variant being close in spirit to decentralized gradient techniques. Section 3.2 is concerned with an alternative approach in which every agent accepts steps solely based on its own function decrease.

3.1 Algorithm based on Lyapunov function decrease

Our first algorithm is based on the penalty formulation (4), and asumes that every agent has access to its own function, neighbor copies of the variable as well as the penalty parameter $\gamma > 0$. The objective function of (4) can be rewritten as

$$\begin{aligned}
\mathcal{L}(\mathbf{x};\gamma) &= \sum_{i=1}^m f_i(x_i) + \frac{1}{2\gamma}\left(\sum_{i=1}^m \|x_i\|^2 - \sum_{i=1}^m \sum_{j=1}^m w_{ij} x_i^{\mathrm{T}} x_j\right) \\
&= \sum_{i=1}^m f_i(x_i) + \frac{1}{2\gamma}\left(\sum_{i=1}^m (1 - w_{ii})\|x_i\|^2 - \sum_{i=1}^m \sum_{j \in \mathcal{N}_i} w_{ij} x_i^{\mathrm{T}} x_j\right).
\end{aligned}$$

When W satisfies Assumption 2.3 (in particular $\lambda_1(W) = 1$), we know that the quantity

$$\sum_{i=1}^m \left(\|y_i\|^2 - \sum_{j=1}^m w_{ij} y_i^{\mathrm{T}} y_j\right) = \sum_{i=1}^m \left((1 - w_{ii})\|y_i\|^2 - \sum_{j \in \mathcal{N}_i} w_{ij} y_i^{\mathrm{T}} y_j\right)$$

is nonnegative for any $y_1, \dots, y_m \in \mathbb{R}^n$ [27]. As a result, under Assumption 2.2, f_{low} is also a lower bound on $\mathcal{L}(\cdot;\gamma)$ for any $\gamma > 0$. Therefore, for a given agent i, we consider a specific local

Lyapunov function defined as

$$\mathcal{L}_i\left(x_i; x_{\mathcal{N}_i}, \gamma\right) := f_i(x_i) + \frac{1}{2\gamma}\left[(1-w_{ii})\|x_i\|^2 - 2\sum_{j\in\mathcal{N}_i} w_{ij} x_i^{\mathrm{T}} x_j\right], \tag{7}$$

where $x_{\mathcal{N}_i} = \{x_j\}_{j\in\mathcal{N}_i}$ represents the information transferred to agent i from its neighbours. With this definition, we have

$$\nabla\mathcal{L}(\mathbf{x};\gamma) = \begin{bmatrix} \nabla_{x_1}\mathcal{L}_1\left(x_1; x_{\mathcal{N}_1}, \gamma\right) \\ \vdots \\ \nabla_{x_m}\mathcal{L}_m\left(x_m; x_{\mathcal{N}_m}, \gamma\right) \end{bmatrix}.$$

During a decentralized optimization process, every agent i updates its local copy by performing an (approximate) minimization step of the function $\mathcal{L}_i(\cdot; x_{\mathcal{N}_i}, \gamma)$, then broadcasts its local copy to its neighbors and receives their local copies. The agent then updates its function before performing another approximate minimization process.

Algorithm 1 describes a direct-search version of this approach, that allows every agent to perform one step of direct-search on this local penalty function at every iteration. The method follows a standard direct-search framework with sufficient decrease, that every agent applies in parallel to its own Lyapunov function $\mathcal{L}_i\left(x_i; x_{\mathcal{N}_i}, \gamma\right)$. Each agent polls a set of directions defined by a positive spanning set $\{D^{(k)}\}$. If there is at least one direction for which the local Lyapunov function $\mathcal{L}_i$ exhibits sufficient decrease, defined by (8), then a step along that direction (scaled by the current stepsize $\alpha^{(k)}$) is taken. Otherwise, this agent's local copy is not updated. Communication here is implicit: evaluation of the conditions for sufficient decrease requires knowledge of the current estimates of $x_{\mathcal{N}_i}^{(k)}$, and thus an iteration of Algorithm 1 requires each agent to communicate its local copy to its neighbors.

3.2 Algorithm based on local function decrease

Our second algorithmic proposal is described in Algorithm 1. It hews closer to standard direct search, in that every agent decides to accept or reject a step based on whether a sufficient decrease condition is satisfied for its own local function. This idea is also in agreement with the DGD principle (5), since the negative gradient of the local function f_i but not necessarily for the penalty function.

In addition to differing from Algorithm 1 in the sufficient decrease acceptance condition, Algorithm 2 also updates the iterate of every agent at each iteration through a consensus step involving the mixing matrix W. This approach is a significant difference with centralized direct-search techniques, and is key to guaranteeing asymptotic consensus between all agents. As we will show in Section 3.2, this seemingly more natural variant is more challenging to analyze.

4 Convergence results

4.1 Generic assumptions

This section details the assumptions that are common to our two algorithms. We will assume, as it is done in classical directional direct-search [14], that all positive spanning sets considered by the algorithm include bounded directions and have cosine measure bounded away from zero.

Algorithm 1: Decentralized direct-search based on local Lyapunov decrease (`DDS-L`)

```
Inputs: Initial points $x_1^{(0)} = \cdots = x_m^{(0)} \in \mathbb{R}^n$ and initial stepsizes $\alpha_1^{(0)} = \cdots = \alpha_m^{(0)} > 0$.
Consensus parameter $\gamma > 0$, sequence of positive spanning sets $\{D^{(k)}\}$, forcing function
$\rho : \mathbb{R}^+ \to \mathbb{R}^+$, mixing matrix $W \in \mathbb{R}^{m \times m}$.
for each iteration $k = 0, 1, 2, 3...$ do
  for each agent $i = 1, \ldots, m$ do
    if there exists $d_i^{(k)} \in D^{(k)}$ such that
        $\mathcal{L}_i\left(x_i^{(k)} + \alpha_i^{(k)} d_i^{(k)}, x_{\mathcal{N}_i}^{(k)}, \gamma\right) \leq \mathcal{L}_i\left(x_i^{(k)}, x_{\mathcal{N}_i}^{(k)}, \gamma\right) - \rho(\alpha_i^{(k)})$    (8)
    then
      Set $x_i^{(k+1)} = x_i^{(k)} + \alpha_i^{(k)} d_i^{(k)}$ and declare the iteration successful for agent $i$.
    else
      Set $x_i^{(k+1)} = x_i^{(k)}$ and declare the iteration unsuccessful for agent $i$.
    end if
    Compute $\alpha_{i+1}^{(k)}$.
  end for
end for
```

Assumption 4.1 *Consider the sequence $\{D^{(k)}\}$ of positive spanning sets used in either Algorithm 1 or 2. There exists $\kappa \in (0,1)$ such that for every k, the set $D^{(k)}$ is a κ-descent set, i.e.,*

$$\mathrm{cm}(D^{(k)}) := \min_{v \neq 0_{\mathbb{R}^n}} \max_{d \in D^{(k)}} \frac{d^{\mathrm{T}} v}{\|d\| \|v\|} \geq \kappa. \tag{10}$$

A consequence of Assumption 4.1 is that $\|d_{(i)}^{(k)}\| = 1$ for $d_{(i)}^{(k)} \in D^{(k)}$, $\forall k$ and $\forall i$. Note that this choice is made for simplicity of exposure, and that the analysis easily generalizes to the case of directions that are uniformly bounded in norm, i.e. when there exist $0 < \beta_{\min} \leq \beta_{\max} < \infty$ such that

$$\forall k, \ \forall d \in D^{(k)}, \quad \beta_{\min} \leq \|d\| \leq \beta_{\max}.$$

A typical choice of direction set that satisfies Assumption 4.1 is $D^{(k)} = D_{\oplus} = [I \ \ -I]$ (in that case $\kappa = \frac{1}{\sqrt{n}}$).

As in standard convergence analyses of direct-search schemes, we rely on the following requirements for the forcing function. Those guarantee in particular that the sufficient decrease condition can be satisfied for a sufficiently small step size [6].

Assumption 4.2 *The forcing function $\rho : \mathbb{R}^+ \to \mathbb{R}^+$ used in either Algorithm 1 or Algorithm 2 satisfies the three properties below.*

(i) *ρ is nondecreasing,*

(ii) *$\alpha \mapsto \frac{\rho(\alpha)}{\alpha}$ is nondecreasing,*

(iii) *$\rho(\alpha) = o(\alpha)$ as $\alpha \to 0$.*

Algorithm 2: Decentralized direct-search based on local function decrease (`DDS-F`)

```
Inputs: Initial points $x_1^{(0)} = \cdots = x_m^{(0)} \in \mathbb{R}^n$ and initial stepsizes $\alpha_1^{(0)} = \cdots = \alpha_m^{(0)} > 0$.
Sequence of positive spanning sets $\{D^{(k)}\}$, forcing function $\rho : \mathbb{R}^+ \to \mathbb{R}^+$, and mixing matrix $W \in \mathbb{R}^{m \times m}$.
for each iteration $k = 0, 1, 2, 3...$ do
  for each agent $i = 1, \ldots, m$ do
    if there exists $d_i^{(k)} \in D^{(k)}$ such that
        $$f_i\left(x_i^{(k)} + \alpha_i^{(k)} d_i^{(k)}\right) \leq f_i\left(x_i^{(k)}\right) - \rho(\alpha_i^{(k)}) \qquad (9)$$
    then
      Set $x_i^{(k+1)} = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} x_j^{(k)} + \alpha_i^{(k)} d_i^{(k)}$ and declare the iteration successful for agent $i$.
    else
      Set $x_i^{(k+1)} = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} x_j^{(k)}$ and declare the iteration unsuccessful for agent $i$.
    end if
    Compute $\alpha_{i+1}^{(k)}$.
  end for
end for
```

Finally, we state our key assumptions on the step size sequences used by our algorithms, that are instrumental to derive theoretical results for our method.

Assumption 4.3 *For the stepsize sequences $\{\alpha_i^{(k)}\}$ used in either Algorithm 1 or Algorithm 2, there exist two sequences $\{\alpha_{\max}^{(k)}\}$ and $\{\alpha_{\min}^{(k)}\}$ such that:*

(i) $\alpha_{\min}^{(k)} \leq \alpha_i^{(k)} \leq \alpha_{\max}^{(k)}$ *for all indices* $(i,k) \in \{1, \ldots, m\} \times \mathbb{N}$*;*

(ii) The sequence $\{\alpha_{\max}^{(k)}\}$ is square summable, i.e.,

$$\sum_{k \in \mathbb{N}} (\alpha_{\max}^{(k)})^2 < \infty. \tag{11}$$

(iii) The sequence $\{\rho(\alpha_{\min}^{(k)})\}$ is not summable, i.e.,

$$\sum_{k \in \mathbb{N}} \rho(\alpha_{\min}^{(k)}) = \infty. \tag{12}$$

A few remarks are in order regarding Assumption 4.3. First, square summability of stepsizes as in (11) is a standard property that is used in both direct-search [9] and decentralized gradient methods with a unique decreasing stepsize [28]. Secondly, property (12) on the forcing function departs from classical choices used in direct-search, such as $\rho(\alpha) = \alpha^2$ [14], while relating to standard requirements in decentralized gradient techniques. Thirdly, we assume properties that involve the maximal and minimal stepsizes among all agents.

A possible choice for satisfying Assumption 4.3 consists in predefining the sequences $\{\alpha_i^{(k)}\}$ independently of the agents . For instance, for any $k \in \mathbb{N}$, one may set

$$\forall i \in \{1, \ldots, m\}, \qquad \alpha_i^{(k)} = \alpha_{\min}^{(k)} = \alpha_{\max}^{(k)} = \frac{\alpha_0}{(1+k)^{\tau_\alpha}} \quad \text{and} \quad \rho(\alpha_i^{(k)}) = \frac{\rho_0}{(1+k)^{\tau_\rho}}, \tag{13}$$

where $\alpha_0 > 0$, $\rho_0 > 0$, $0.5 < \tau_\alpha < \tau_\rho \leq 1$. It is clear that the resulting sequences satisfy Assumption 4.3. This choice is in line with the standard practice in decentralized optimization, that favors a priori stepsize rules [19].

An alternate stepsize choice, closer to that of direct-search schemes, consists in updating $\alpha_i^{(k)}$ in an adaptive fashion for every agent. More precisely, if sufficient decrease holds for agent i at iteration k, then iteration k is successful and one possibly increases $\alpha_i^{(k)}$. Otherwise, the iteration is unsuccessful, in which case one decreases $\alpha_i^{(k)}$. In addition, the choice of ρ is made so as to satisfy Assumption 4.3. Overall, for any $i \in \{1, \ldots, m\}$ and $k \in \mathbb{N}$, the update formulas are given by

$$\alpha_i^{(k)} = \begin{cases} \min\left\{\theta^{-1}\alpha_i^{(k)}, \alpha_{\max}^{(k)}\right\} & \text{if } k \text{ is successful for } i \\ \max\left\{\theta\alpha_i^{(k)}, \alpha_{\min}^{(k)}\right\} & \text{otherwise,} \end{cases} \qquad \text{and} \quad \rho(\alpha_i^{(k)}) = c\left(\alpha_i^{(k)}\right)^{1+\tau_\rho}, \tag{14}$$

where $\{\alpha_{\max}^{(k)}\}$ and $\{\alpha_{\min}^{(k)}\}$ are positive sequences that converge towards 0, $\theta \in (0,1)$, $\tau_\rho \in (0,1)$, and $c > 0$.

The analysis in the upcoming sections will rely heavily on Assumption 4.3 while distinguishing between successful iterations (for which at least one agent updates its local copy) and unsuccessful iterations (for which all agents leave their local copies unchanged). To this end, we let $\mathcal{S}^{(k)} \subseteq \{1, \ldots, m\}$ (resp. $\mathcal{U}^{(k)} \subseteq \{1, \ldots, m\}$) denote the set of agents for which iteration k is successful (resp. unsuccessful).

4.2 Convergence analysis of Algorithm 1 (`DDS-L`)

We begin by analyzing Algorithm 1. Since this method relies on a penalty function defined with a constant penalty parameter, it cannot be expected to converge to a solution of problem (2), in the sense that consensus is not guaranteed. However, we will show that the method converges towards a stationary point for problem (4), akin to decentralized gradient schemes [28].

An iteration of Algorithm 1 corresponds to applying one step of a direct-search algorithm to the function $\mathcal{L}_i$ for agent i. Classical analyses of direct-search methods rely on a link between the gradient of the objective and the stepsize on unsuccessful iterations. The following lemma provides an analogous result for the decentralized setting.

Lemma 4.1 *Let Assumptions 2.1 and 4.1 hold. Suppose that the k-th iteration of Algorithm 1 is unsuccessful for agent i. Then, one has*

$$\left\|\nabla_{x_i}\mathcal{L}_i\left(x_i^{(k)}; x_{\mathcal{N}_i}^{(k)}, \gamma\right)\right\| \leq \frac{1}{\kappa}\left(\frac{M_i}{2}\alpha_i^{(k)} + \frac{\rho(\alpha_i^{(k)})}{\alpha_i^{(k)}}\right), \tag{15}$$

where $M_i := L_i + \frac{1-w_{ii}}{\gamma}$.

Proof. Since the k-th iteration is unsuccessful for agent i, condition (8) does not hold. Therefore, we must have

$$\mathcal{L}_i\left(x_i^{(k)}; x_{\mathcal{N}_i}^{(k)}, \gamma\right) - \rho(\alpha_i^{(k)}) < \mathcal{L}_i\left(x_i^{(k)} + \alpha_i^{(k)} d; x_{\mathcal{N}_i}^{(k)}, \gamma\right) \tag{16}$$

for every $d \in D^{(k)}$. In particular, letting $g_i^{(k)} = \nabla_{x_i}\mathcal{L}_i\left(x_i^{(k)}; x_{\mathcal{N}_i}^{(k)}, \gamma\right)$ and $\bar{d}_i := \arg\max_{d \in D^{(k)}} \frac{d^{\mathrm{T}}\left[-g_i^{(k)}\right]}{\|d\|\left\|g_i^{(k)}\right\|}$, we have by Assumption 4.1 that $\bar{d}_i^{\mathrm{T}}\left[g_i^{(k)}\right] \le -\kappa\|g_i^{(k)}\|$.

Now, by Assumption 2.1, the function $\mathcal{L}_i(\cdot; x_{\mathcal{N}_i}^{(k)}, \gamma)$ is continuously differentiable, and its gradient is Lipschitz continuous with Lipschitz constant $M_i^{(k)} := L_i + \frac{1-w_{ii}}{\gamma}$. As a result,

$$\begin{aligned}
\mathcal{L}_i\left(x_i^{(k)} + \alpha^{(k)}\bar{d}_i; x_{\mathcal{N}_i}^{(k)}, \gamma\right) &\le \mathcal{L}_i\left(x_i^{(k)}; x_{\mathcal{N}_i}^{(k)}, \gamma\right) + \alpha_i^{(k)}\bar{d}_i^{\mathrm{T}}\left[g_i^{(k)}\right] + \frac{M_i}{2}(\alpha_i^{(k)})^2 \\
&\le \mathcal{L}_i\left(x_i^{(k)}; x_{\mathcal{N}_i}^{(k)}, \gamma\right) - \alpha_i^{(k)}\kappa\|g_i^{(k)}\| + \frac{M_i}{2}(\alpha_i^{(k)})^2.
\end{aligned}$$

Combining the last inequality with (16) applied at $\bar{d}$ leads to

$$\begin{aligned}
-\rho(\alpha_i^{(k)}) &\le \mathcal{L}_i\left(x_i^{(k)} + \alpha_i^{(k)}\bar{d}_i; x_{\mathcal{N}_i}^{(k)}, \gamma\right) - \mathcal{L}_i\left(x_i^{(k)}; x_{\mathcal{N}_i}^{(k)}, \gamma\right) \\
&\le -\alpha_i^{(k)}\kappa\|g_i^{(k)}\| + \frac{M_i}{2}(\alpha_i^{(k)})^2.
\end{aligned}$$

Re-arranging the terms and replacing $\alpha_i^{(k)}$ and $\beta_i^{(k)}$ by their expressions, we obtain:

$$\|g_i^{(k)}\| \le \frac{1}{\kappa}\left[\frac{M_i}{2}\alpha_i^{(k)} + \frac{\rho(\alpha_i^{(k)})}{\alpha_i^{(k)}}\right],$$

proving the desired result. ■

The contrapositive of Lemma 4.1 implies that an iteration for which (15) does not hold is necessarily successful. In centralized direct-search, this property is combined with the fact that the stepsize sequence goes to zero to produce convergence and complexity guarantees [26]. In our case, our assumptions on the stepsize and forcing function sequences yields the following result.

Theorem 4.1 *Let Assumptions 2.3, 4.1, 2.1, 2.2 and 4.3 hold. Suppose further that* $\{\alpha_i^{(k)}\}_k \to 0$ *for any* $i \in \{1, \dots, m\}$*. Then, the sequence of iterates* $\left\{\{x_i^{(k)}\}_{i=1}^m\right\}_k$ *generated by Algorithm 1 satisfies*

$$\liminf_{k\to\infty} \left\|\nabla\mathcal{L}(\mathbf{x}^{(k)}, \gamma)\right\| = 0. \tag{17}$$

Proof. We proceed by contradiction. Suppose that $\|\nabla\mathcal{L}(\mathbf{x}^{(j)}, \gamma)\| > \epsilon$ for any $j \in \mathbb{N}$. Then, by definition of this gradient, this implies

$$\sum_{i=1}^m \left\|\nabla\mathcal{L}_i(x_i^{(j)}; x_{\mathcal{N}_i}^{(j)}, \gamma)\right\| \ge \epsilon \quad \forall j \in \mathbb{N},$$

hence there must exist an agent i_j such that

$$\left\|\nabla \mathcal{L}_{i_j}(x_{i_j}^{(j)}; x_{\mathcal{N}_{i_j}}^{(j)}, \gamma)\right\| \geq \frac{\epsilon}{m},$$

Since $\alpha_{\max}^{(k)} \to 0$, we now that there exists K_ϵ such that for every $k \geq K_\epsilon$, we have

$$\max_{1\leq i\leq m} \alpha_i^{(k)} \leq \alpha_{\max}^{(k)} < \inf\left\{ \alpha > 0 \;\middle|\; \frac{\epsilon}{m} \leq \frac{1}{\kappa}\left(\frac{\min_{1\leq i\leq m} M_i}{2}\alpha + \frac{\rho(\alpha)}{\alpha}\right) \right\}, \tag{18}$$

and therefore the k-th iteration will be successful for at least one agent. Overall, assuming that (17) does not hold, there exists K_ϵ such that every iteration of index $k \geq K_\epsilon$ is successful for at least one agent.

For any $k \geq K_\epsilon$, a Taylor expansion of $\mathcal{L}(\cdot, \gamma)$ gives

$$\mathcal{L}(\mathbf{x}^{(k+1)}; \gamma) - \mathcal{L}(\mathbf{x}^k; \gamma) \quad\leq\quad \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^{(k)}; \gamma)^{\mathrm{T}}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) + \frac{M}{2}\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|^2,$$

where $M = \max_{1\leq i\leq m} M_i$. Recalling that $\mathbf{x}^{(k)} = [x_i^{(k)}]_{i=1}^m$ and

$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^{(k)}; \gamma) = [\nabla_{x_i} \mathcal{L}(x_i^{(k)}; x_{\mathcal{N}_i}^{(k)}, \gamma)]_{i=1}^m$ for every k, we obtain:

$$\begin{aligned}
\mathcal{L}(\mathbf{x}^{(k+1)}; \gamma) - \mathcal{L}(\mathbf{x}^{(k)}; \gamma) &\leq \sum_{i=1}^m \left[\nabla \mathcal{L}_i(x_i^{(k)}; x_{\mathcal{N}_i}^{(k)}, \gamma)^{\mathrm{T}}(x_i^{(k+1)} - x_i^{(k)}) + \frac{M}{2}\|x_i^{(k+1)} - x_i^{(k)}\|^2\right] \\
&= \sum_{i\in\mathcal{S}^{(k)}} \left[\nabla \mathcal{L}_i(x_i^{(k)}; x_{\mathcal{N}_i}^{(k)}, \gamma)^{\mathrm{T}}(x_i^{(k+1)} - x_i^{(k)}) + \frac{M}{2}\|x_i^{(k+1)} - x_i^{(k)}\|^2\right].
\end{aligned} \tag{19}$$

Using now a Taylor expansion of $\mathcal{L}_i(x_i^{(k+1)}; x_{\mathcal{N}_i}^{(k)}, \gamma)$ for every $i \in \mathcal{S}^{(k)}$, we have:

$$\mathcal{L}_i(x_i^{(k+1)}; x_{\mathcal{N}_i}^{(k)}, \gamma) \geq \mathcal{L}_i(x_i^{(k)}; x_{\mathcal{N}_i}^{(k)}, \gamma) + \nabla \mathcal{L}_i(x_i^{(k)}; x_{\mathcal{N}_i}^{(k)}, \gamma)^{\mathrm{T}}(x_i^{(k+1)} - x_i^{(k)}) - \frac{M_i}{2}\|x_i^{(k+1)} - x_i^{(k)}\|^2,$$

leading to

$$\begin{aligned}
\nabla \mathcal{L}_i(x_i^{(k)}; x_{\mathcal{N}_i}^{(k)}, \gamma)^{\mathrm{T}}(x_i^{(k+1)} - x_i^{(k)}) &\leq \mathcal{L}_i(x_i^{(k+1)}; x_{\mathcal{N}_i}^{(k)}, \gamma) - \mathcal{L}_i(x_i^{(k)}; x_{\mathcal{N}_i}^{(k)}, \gamma) + \frac{M_i}{2}\|x_i^{(k+1)} - x_i^{(k)}\|^2 \\
&\leq -\rho(\alpha_i^{(k)}) + \frac{M_i}{2}\|x_i^{(k+1)} - x_i^{(k)}\|^2,
\end{aligned} \tag{20}$$

since the iteration is successful for agent i. Plugging (20) into (19) then gives

$$\begin{aligned}
\mathcal{L}(\mathbf{x}^{(k+1)}; \gamma) - \mathcal{L}(\mathbf{x}^{(k)}; \gamma) &\leq \sum_{i\in\mathcal{S}^{(k)}} \left[-\rho(\alpha_i^{(k)}) + \frac{M + M_i}{2}\|x_i^{(k+1)} - x_i^{(k)}\|^2\right] \\
&= \sum_{i\in\mathcal{S}^{(k)}} \left[-\rho(\alpha_i^{(k)}) + \frac{M + M_i}{2}(\alpha_i^{(k)})^2\right] \\
&\leq \sum_{i\in\mathcal{S}^{(k)}} \left[-\rho(\alpha_i^{(k)}) + M(\alpha_i^{(k)})^2\right] \\
&\leq \sum_{i\in\mathcal{S}^{(k)}} \left[-\rho(\alpha_{\min}^{(k)}) + M(\alpha_{\max}^{(k)})^2\right] \\
\mathcal{L}(\mathbf{x}^{(k+1)}; \gamma) - \mathcal{L}(\mathbf{x}^{(k)}; \gamma) &\leq -\rho(\alpha_{\min}^{(k)}) + mM(\alpha_{\max}^{(k)})^2,
\end{aligned} \tag{21}$$

where the last inequality comes from the fact that the iteration is successful for at least one agent, and at most for all m agents.

Now, for any $J > K_\epsilon$, summing (21), for any $k \in \{K_\epsilon, \dots, J-1\}$, gives

$$\begin{aligned} \sum_{k=K_\epsilon}^{J-1} \rho(\alpha_{\min}^{(k)}) &\leq \mathcal{L}(\mathbf{x}^{(K_\epsilon)}; \gamma) - \mathcal{L}(\mathbf{x}^{(J)}; \gamma) + mM \sum_{k=K_\epsilon}^{J-1} (\alpha_{\max}^{(k)})^2 \\ &\leq \mathcal{L}(\mathbf{x}^{(K_\epsilon)}; \gamma) - f_{\text{low}} + mM \sum_{k=K_\epsilon}^{J-1} (\alpha_{\max}^{(k)})^2, \end{aligned}$$

where the last inequality comes from Assumption 2.2. As $J \to +\infty$, the left-hand side goes to $+\infty$ while the right-hand side is finite by Assumption 4.3. Therefore, we obtain a contradiction, from which we conclude that (17) holds. ■

Theorem 4.1 shows that Algorithm 1 converges to a stationary point of the penalized objective (4), but does not provide consensus guarantees. As we will show in our experiments, consensus for this approach does improve as the penalty parameter γ decreases.

4.3 Convergence analysis of Algorithm 2 (DDS-F)

We now turn to Algorithm 2, whose analysis requires a number of ingredients from the decentralized optimization literature. On one hand, Assumption 4.3 is critical for convergence as it guarantees convergence of the stepsize sequence to zero, akin to standard analyses of decentralized gradient descent in a nonconvex setting [27]. On the other hand, the properties of the mixing matrix W are instrumental for reaching asymptotic consensus, as they yield the following guarantees.

Proposition 4.1 *[19, Proposition 1] Under Assumption 2.3, there exists a constant $C_W \geq 0$ such that*

$$\left\| \widehat{W}^j - A_m \right\| \leq C_W \zeta^k$$

for any $j \in \mathbb{N}$, where $A_m := \frac{1}{m}\left(\mathbf{1}_m \mathbf{1}_m^{\mathrm{T}}\right) \otimes \mathbf{I}_n$ and ζ is the spectral mixing matrix constant defined by

$$\zeta := \max\left(|\lambda_2(W)|, |\lambda_n(W)|\right). \tag{22}$$

Lemma 4.2 *[28, Lemma 8] Under Assumptions 2.3 and 4.3, there exists $C_\zeta \geq 0$ such that*

$$\sum_{j=0}^{k} \zeta^{k-j} \alpha_{\max}^{(j)} \leq C_\zeta \alpha_{\max}^{(k)}$$

for any $k \in \mathbb{N}$.

The previous results allow to bound the consensus error at every iteration, by a reasoning similar to the derivative-based setting [28, Proposition 3]. Since our approach does not necessarily update each local copy at every iteration, we provide a full proof of that result.

Proposition 4.2 *Let Assumption 2.3 hold. For any $k \in \mathbb{N}$, we have*

$$\left\| \mathbf{x}^{(k)} - A_m \mathbf{x}^{(k)} \right\| \leq \sqrt{m} C_W \left(\|\mathbf{x}^0\| \zeta^k + C_\zeta \alpha_{\max}^{(k-1)} \right)$$

where C_W and C_ζ are the constants defined in Proposition 4.1 and Lemma 4.2, respectively.

Proof. By definition of the kth iteration, we have

$$\mathbf{x}^{(k)} = \widehat{W}^k \mathbf{x}^{(0)} + \sum_{j=0}^{k-1} \widehat{W}^{k-1-j} \left(\alpha^{(j)} \otimes \mathbf{1}_n\right) \odot \mathbf{d}^{(j)},$$

where $\mathbf{d}^{(j)} \in \mathbb{R}^{mn}$ concatenates the directions used by each agent at iteration j, i.e. $d_i^{(j)}$ if $i \in \mathcal{S}^{(j)}$ and $0_{\mathbb{R}^n}$ otherwise. As a result,

$$\mathbf{x}^{(k)} - A_m\,\mathbf{x}^{(k)} \;=\; \left(\widehat{W}^k - A_m\widehat{W}^k\right)\mathbf{x}^{(0)} + \sum_{j=0}^{k-1}\left(\widehat{W}^{k-1-j} - A_m\widehat{W}^{k-1-j}\right)\left(\alpha^{(j)} \otimes \mathbf{1}_n\right) \odot \mathbf{d}^{(j)}.$$

Meanwhile, by Assumption 2.3, we have $A_m\widehat{W}^{k-1-j}\mathbf{v} = A_m\mathbf{v}$ for any $\mathbf{v} \in \mathbb{R}^{nm}$ and any $j \in \{0, \ldots, k-1\}$. Using this property together with Cauchy-Schwarz inequality, we obtain

$$\left\|\mathbf{x}^{(k)} - A_m\mathbf{x}^{(k)}\right\| \;\le\; \left\|\widehat{W}^k - A_m\right\| \|\mathbf{x}^{(0)}\| + \sum_{j=0}^{k-1} \|\widehat{W}^{k-1-j} - A_m\| \left\|\left(\alpha^{(j)} \otimes \mathbf{1}_n\right) \odot \mathbf{d}^{(j)}\right\|.$$

For any $j \in \{0, \ldots, k-1\}$, we have

$$\left\|\left(\alpha^{(j)} \otimes \mathbf{1}_n\right) \odot \mathbf{d}^{(j)}\right\| = \sqrt{\sum_{i \in \mathcal{S}^{(j)}} \left(\alpha_i^{(j)}\|d_i^{(j)}\|\right)^2} \le \sqrt{m}\alpha_{\max}^{(j)},$$

where we used that $\|\mathbf{d}^{(j)}\|^2 = \sum_{i\in\mathcal{S}^{(j)}} \|d_i^{(j)}\|^2$ and $\|d_i^{(j)}\| = 1$. Thus,

$$\begin{aligned}\left\|\mathbf{x}^{(k)} - A_m\mathbf{x}^{(k)}\right\| \;&\le\; \left\|\widehat{W}^k - A_m\right\| \|\mathbf{x}^{(0)}\| + \sum_{j=0}^{k-1} \|\widehat{W}^{k-1-j} - A_m\|\sqrt{m}\alpha_{\max}^{(j)} \\ &\le\; C_W\zeta^k\|\mathbf{x}^{(0)}\| + \sum_{j=0}^{k-1} \sqrt{m}C_W\zeta^{k-1-j}\alpha_{\max}^{(j)} \\ &\le\; C_W\|\mathbf{x}^{(0)}\|\zeta^k + C_W\sqrt{m}\alpha_{\max}^{(k-1)}\end{aligned}$$

where the second inequality follows from Proposition 4.1 and the third one follows from Lemma 4.2. Rearranging the constants yields the desired conclusion. ■

Provided the stepsize is chosen so as to satisfy our assumptions, one can establish that the iterates of Algorithm 2 reach asymptotic consensus.

Theorem 4.2 *Under Assumptions 2.3 and 4.3, the iterates of Algorithm 2 satisfy*

$$\lim_{k\to\infty} \left\|\mathbf{x}^{(k)} - \widehat{W}\mathbf{x}^{(k)}\right\| = 0.$$

Proof. For any $k \in \mathbb{N}$, we have

$$\|\mathbf{x}^{(k)} - \widehat{W}\mathbf{x}^{(k)}\| \le \|\mathbf{x}^{(k)} - A_m\mathbf{x}^{(k)}\| + \|A_m\mathbf{x}^{(k)} - \widehat{W}\mathbf{x}^{(k)}\|.$$

Noticing that

$$A_m \widehat{W}\mathbf{x}^{(k)} = A_m \mathbf{x}^{(k)} = \widehat{W} A_m \mathbf{x}^{(k)},$$

we obtain

$$\begin{aligned} \|\mathbf{x}^{(k)} - \widehat{W}\mathbf{x}^{(k)}\| &= \|\mathbf{x}^{(k)} - A_m\mathbf{x}^{(k)}\| + \|\widehat{W}A_m\mathbf{x}^{(k)} - \widehat{W}\mathbf{x}^{(k)}\| \\ &\leq (1+\|\widehat{W}\|)\|\mathbf{x}^{(k)} - A_m\mathbf{x}^{(k)}\| \\ &\leq \sqrt{m}C_W(1+\|\widehat{W}\|)\left(\|\mathbf{x}^0\|\zeta^k + C_\zeta \alpha_{\max}^{(k-1)}\right), \end{aligned} \tag{23}$$

where the last inequality is from Proposition 4.2. By Assumption 2.3, $\zeta \in (0,1)$ and thus $\lim_{k\to\infty}\zeta^k = 0$. By Assumption 4.3, $\sum_k (\alpha_{\max}^{(k)})^2 < \infty$ and therefore

$$\lim_{k\to\infty} \alpha_{\max}^{(k)} = \lim_{k\to\infty} \alpha_{\max}^{(k-1)} = 0.$$

Combining these observations with (23) yields the desired conclusion. ■

The result of Theorem 4.2 is somewhat expected, since Algorithm 2 performs a consensus step at every iteration for all agents, regardless of the successful nature of that iteration. Note however that Assumption 4.3 (and in particular the fact that $\{\alpha_{\max}^{(k)}\}$ converges to zero) is instrumental to such a result.

Unlike DGD techniques, however, we cannot establish convergence to a first-order stationary point for this framework, in part because the decrease condition may not be in line with the consensus step, while relying on directions that are not directly related to the true gradients. We illustrate this issue using a two-dimensional example with two agents. Suppose that $n = m = 2$ and that

$$f_1(x) = (x_1 - 1)^2, \quad f_2(x) = x_2^2 \quad \text{and} \quad W = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Suppose further that we apply Algorithm 2 with $x_1^{(0)} = x_2^{(0)} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ and $D_k = D = \{d, -d, \dots\}$ with $d = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$. Then, any decreasing stepsize sequence $\alpha^{(k)}$ such that the iteration is always successful for agent 1 using direction d and for agent 2 using direction $-d$ would lead to consensus without optimality. Indeed, in such a case, the average iterate would always equal $\begin{bmatrix} 0 & 1 \end{bmatrix}^T$, while the iterates would have the form

$$x_1^{(k)} = \begin{bmatrix} \alpha_k \\ 1+\alpha_k \end{bmatrix}, \quad x_2^{(k)} = \begin{bmatrix} -\alpha_k \\ 1-\alpha_k \end{bmatrix},$$

implying that Algorithm 2 never converges to a minimizer. This example cast doubts on the convergence guarantees of Algorithm 2, yet we have found that method to perform quite well in our experiments, described in the next section.

5 Numerical Results

In this section, we evaluate the performance of our proposed direct-search algorithms compared to zeroth-order variants of decentralized gradient descent. We first compare our algorithms on a toy problem from the decentralized literature [11]. We then adapt the Moré-Wild test set [18], a standard benchmark in derivative-free optimization, to the decentralized setting.

5.1 Implementation details

Our experiments are conducted in MATLAB version R2024a. Algorithms 1 (DDS-L) and 2 (DDS-F) were implemented using the same positive spanning set across all iterations, namely $D = [B_\oplus, -B_\oplus]$ where $B_\oplus$ corresponds to the canonical basis of $\mathbb{R}^n$. For each method, we considered two variants based on the following stepsize updating rules:

- Vanishing: $\alpha^{(k)} = \alpha^{(k)}_{\max} = \alpha^{(k)}_{\min} = \frac{\alpha^0}{(1+k)^{0.6}}$, $c = 10^{-8}$, and $\tau_\rho = 0.8$;
- Adaptive: $\alpha^{(k)}_{\max} = +\infty$, $\alpha^{(k)}_{\min} = -\infty$, $\theta = 0.5$, $c = 10^{-8}$, and $\tau_\rho = 0.8$.

We compared DDS-L and DDS-F with two zeroth-order decentralized gradient schemes based on the iteration

$$x_i^{(k+1)} = \sum_{j \in \mathcal{N}_i} w_{ij} x_j^{(k)} - \alpha^{(k)} \tilde{g}_i^{(k)} \qquad \forall k \in \mathbb{N},\ \forall i \in \{1, \dots, m\},$$

where $\tilde{g}_i^{(k)}$ is a gradient approximation built from function values. The variant ZO-DGD(FD) is built from centered finite differences based on evaluating f_i along all directions in D with a finite difference parameter of 10^{-3}. The variant ZO-DGD(LM) fits a linear model to previously available values, in the spirit of model-based derivative-free optimization [2].

We run all solvers using $\alpha^0 = \|x_0\| + 1$ as an initial stepsize, where x_0 is the same starting point for all solvers. The underlying network for all problems is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of m agents where agents are connected with probability $p_c = 0.5$.

At each iteration k, we record the following metrics:

- $\sum_{i=1}^m f_i(x^{(k)}_{(i)})$, where $x^{(k)}_{(i)}$ is the iterate associated with agent i at iteration k,
- $\sum_{i=1}^m f_i(\bar{x}^{(k)})$, where $\bar{x}^{(k)} := \frac{1}{m}\sum_{i=1}^m x^{(k)}_{(i)}$ is the average of the iterates associated with all agents m at iteration k,
- $\sum_{i=1}^m \|x^{(k)}_{(i)} - \bar{x}^{(k)}\|$, representing the consensus of all m agents at iteration k.

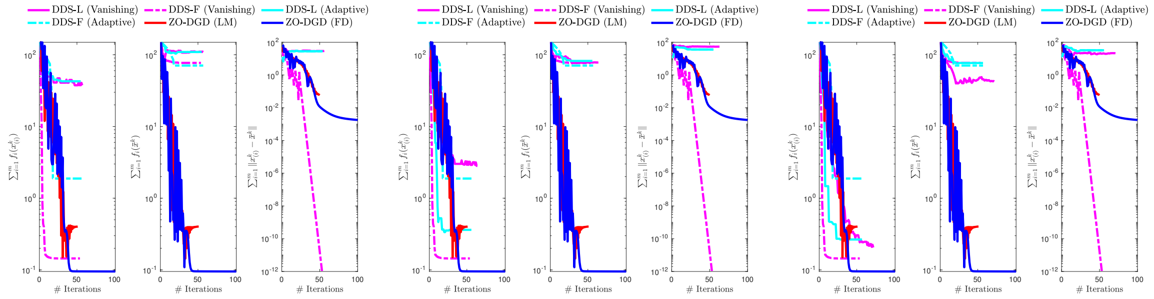
The first two metrics are associated with optimality, while the last metric measures the agreement between the local iterates.

5.2 A separable problem

We consider the following optimization problem from Hajinezhad et al [11]:

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^n f_i(x_i) = \frac{a_i}{1 + \exp(-x_i)} + b_i \log(1 + x_i^2) \qquad x_i = x_j \quad \forall (i,j) \in \mathcal{V}. \tag{24}$$

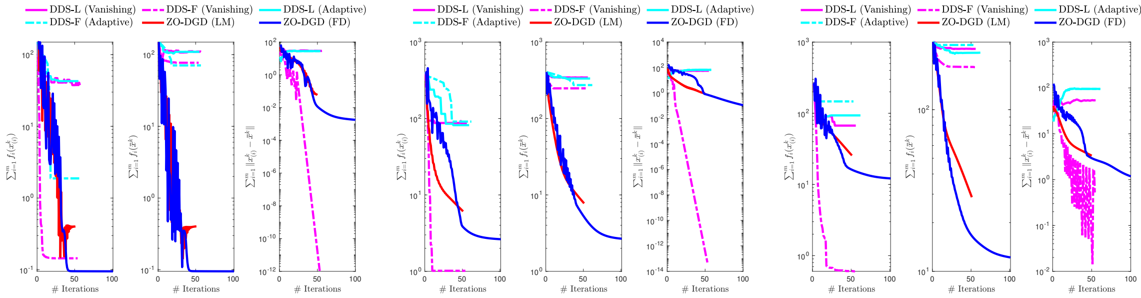
Problem (24) fits our general framework (1) with $m = n$. Moreover, note that the problem is *separable*, in that every function f_i depends solely on the ith optimization variable. We consider three instances of problem (24) corresponding to $n \in \{5, 10, 15\}$. In each instance, the parameters $\{a_i, b_i\}_{i=1,\dots,m}$ were generated following an i.i.d. standard Gaussian distribution. All methods were run with a maximum budget of $100n$ evaluations, using the vector of all ones in $\mathbb{R}^n$ as a starting point.



(a) $\gamma = 1$ (b) $\gamma = 10$ (c) $\gamma = 100$

Figure 1: Convergence plots for problem (24) in dimension $n = 5$.

Figure 1 focuses on the case $n = 5$. Our goal is both to compare our algorithms with zeroth-order schemes, and to investigate the impact of γ on the performance of DDS-L (Algorithm 2). First, note that the best variant in terms of objective value at the iterates and the averaged iterate is the finite-difference variant ZO-DGD (FD). However, we note that DDS-F with vanishing stepsizes converges more quickly in terms of objective value, even though it plateaus at a higher value overall. Besides, the DDS-F (Vanishing) variant outperforms the other methods in terms of consensus, which is a key aspect in decentralized algorithms. We note that increasing γ improves consensus for the DDS-L variants, although those variants are outperformed by the zeroth-order schemes as well as the DDS-F ones.



(a) $n = m = 5$ (b) $n = m = 10$ (c) $n = m = 15$

Figure 2: Convergence plots for problem (24) with $\gamma = 1$.

Figure 2 illustrates the variability in performance as the dimension of the problem changes. We observe again that DDS-F (Vanishing) outperforms the other variants in terms of objective and consensus values for most values, while the finite- difference zeroth-order scheme yields the lowest average objective value. Overall, these results suggest that classical direct-search approaches can outperform zeroth-order schemes in a decentralized setting, akin to the centralized case.

5.3 Decentralized Moré-Wild test set

We now compare our algorithms on a test set built from that of Moré and Wild [18]. This test set comprises 22 nonlinear smooth vector functions of the form $F : \mathbb{R}^n \to \mathbb{R}^m$ where $2 \leq n \leq 30$ and $2 \leq m \leq 65$. In our experiments, we consider that the components of F are aggregated as a sum of squares, yielding an objective of the form $\sum_{i=1}^{m} F_i(x)^2$. The local function of agent i is thus the function $f_i : x \mapsto F_i(x)^2$. We run our algorithms using the default starting points of the test set [18] as well as a maximum budget of either $400nm$ local function evaluations (i.e., a budget of $400n$ evaluations per agent) or 500 iterations.

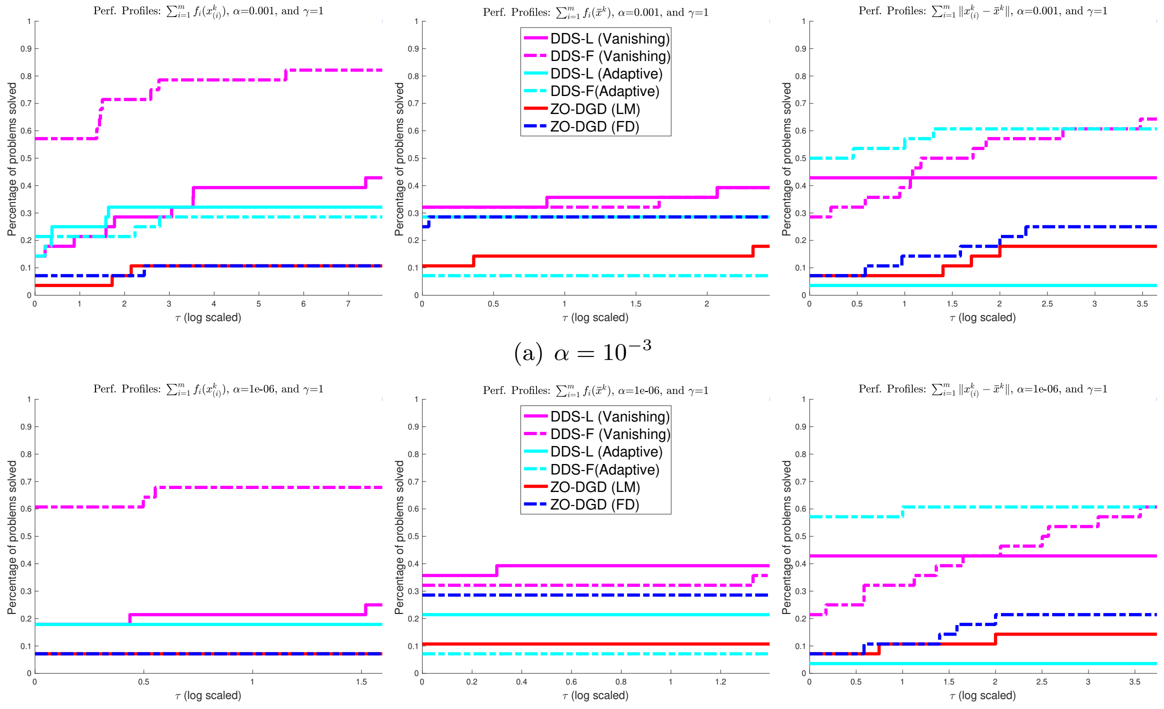
The computational analysis is carried out by using well-known tools from the literature, that is performance and data profiles (see [5, 18] for further details). We briefly recall here their definitions. Given a set $\mathcal{S}$ of algorithms and a set $\mathcal{P}$ of problems, for $s \in \mathcal{S}$ and $p \in \mathcal{P}$, let $t_{p,s}$ be the number of function evaluations required by algorithm s on problem p to satisfy the condition

$$\texttt{opt}(\mathbf{x}^{(k)}) \le \texttt{opt}_{\text{low}} + \alpha \left(\texttt{opt}(\mathbf{x}^{(k)}) - \texttt{opt}_{\text{low}} \right) ,$$

where $\alpha \in (0,1)$, $\texttt{opt}(\mathbf{x}^{(k)})$ is the optimality metric (i.e., $\sum_{i=1}^{m} f_i(x_{(i)}^{(k)})$, $\sum_{i=1}^{m} f_i(\bar{x}^{(k)})$, or $\sum_{i=1}^{m} \|x_{(i)}^{(k)} - \bar{x}^{(k)}\|$), and $\texttt{opt}_{\text{low}}$ is the best optimality metric value achieved by any solver on problem p. Then, the performance and data profiles of solver s are defined by

$$\begin{aligned} \rho_s(\gamma) &:= \frac{1}{|\mathcal{P}|} \left| \left\{ p \in \mathcal{P} : \frac{t_{p,s}}{\min\{t_{p,s'} : s' \in \mathcal{S}\}} \le \gamma \right\} \right|, \\ d_s(\kappa) &:= \frac{1}{|\mathcal{P}|} \left| \{ p \in \mathcal{P} : t_{p,s} \le \kappa(n_p + 1) \} \right| , \end{aligned}$$

where n_p is the dimension of problem p. In our tests, for both data and performance profiles, we used two tolerance choices for α: 10^{-3} and 10^{-6}.

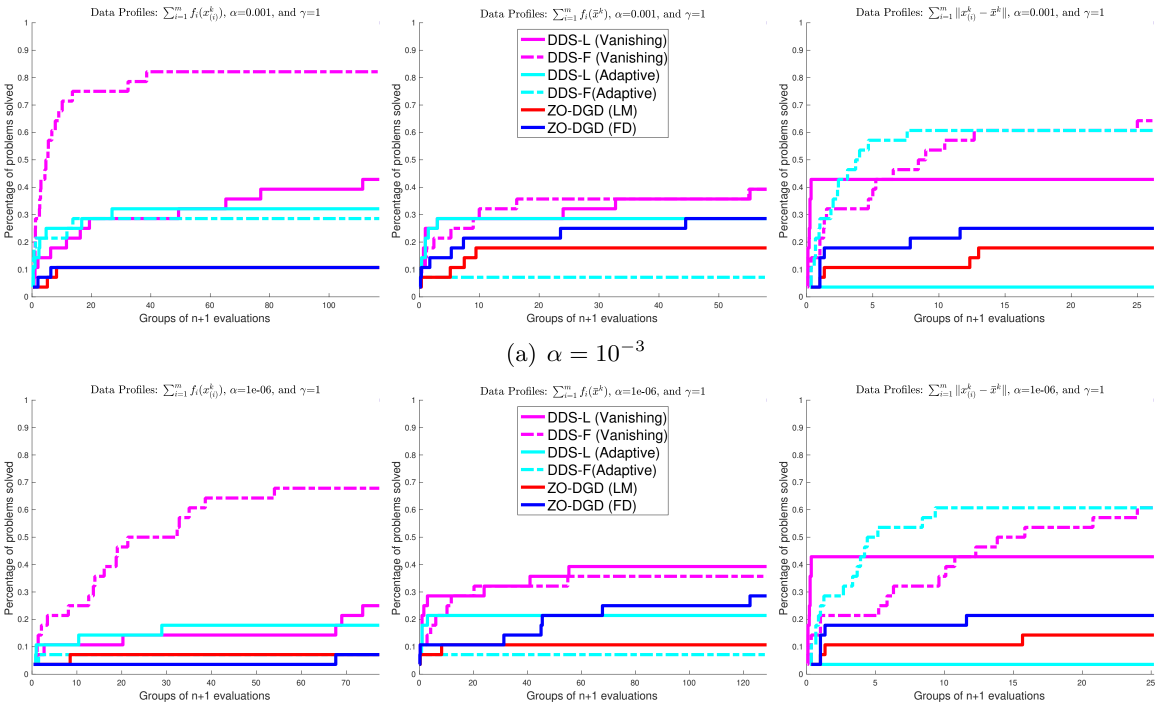


(a) $\alpha = 10^{-3}$

(b) $\alpha = 10^{-6}$

Figure 3: Performance profiles using three different optimality metrics.

Figure 3 shows performance profiles [5] comparing our various algorithms. We observe that the DDS variants mostly outperform the ZO-DGD methods in terms of function values and consensus, with DDS-F (Vanishing) standing out as the best variant overall. We note that the discrepancy between direct-search and zeroth-order methods is less pronounced in terms of function values at the average iterates, which is on par with the observations of Section 5.2.



(a) $\alpha = 10^{-3}$

(b) $\alpha = 10^{-6}$

Figure 4: Data profiles using three different optimality metrics.

Figure 4 complements our study by presenting data profiles [18] for our runs. Those profiles are consistent with the performance profiles, and further illustrate that DDS-F (Vanishing) reaches the best compromise between function value and consensus. Overall, these experiments support the use of direct-search techniques in a decentralized setting.

6 Conclusion

In this paper, we adapted direct-search techniques to operate in a decentralized setting. We proposed sufficient decrease conditions and stepsize updating techniques which borrow from the decentralized gradient descent literature as well asthe derivative-free optimization literature. While only endowed with partial convergence guarantees, our algorithms can outperform zeroth-order decentralized gradient descent techniques in practice

Our study can be extended in several directions. First, other decentralized schemes, such as gradient tracking algorithms, could be combined with direct-search techniques. Extending our framework to account for nonsmoothness or stochasticity in the objective values is also an interesting avenue for future research.

References

[1] C. Audet and W. Hare. *Derivative-Free and Blackbox Optimization.* Springer Series in Operations Research and Financial Engineering. Springer, Cham, Switzerland, 2017.

[2] A.R. Conn, K. Scheinberg, and L.N. Vicente. *Introduction to Derivative-Free Optimization.* MOS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.

[3] Q. Dang, S. Yang, Q. Liu, and J. Ruan. Adaptive and communication-efficient zeroth-order optimization for distributed internet of things. *IEEE Internet of Things Journal*, 11(22):37200–37213, 2024.

[4] P. Di Lorenzo and G. Scutari. NEXT: In-network nonconvex optimization. *IEEE Trans. Signal Inform. Process. Netw.*, 2(2):120–136, 2016.

[5] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91:201–213, 2002.

[6] K. J. Dzahini, F. Rinaldi, C. W. Royer, and D. Zeffiro. Revisiting theoretical guarantees of direct-search methods. arXiv:2403.05322v2, 2024.

[7] H. Gao, M. T. Thai, and J. Wu. When decentralized optimization meets federated learning. *IEEE Network*, 37(5):233–239, 2023.

[8] S. Ghadimi and G. Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM J. Optim.*, 23(4):2341–2368, 2013.

[9] S. Gratton, C. W. Royer, L. N. Vicente, and Z. Zhang. Direct search based on probabilistic descent. *SIAM J. Optim.*, 25(3):1515–1541, 2015.

[10] J. D. Griffin, T. G. Kolda, and R. M. Lewis. Asynchronous parallel generating set search for linearly constrained optimization. *SIAM J. Sci. Comput.*, 30:1892–1924, 2008.

[11] D. Hajinezhad, M. Hong, and A. Garcia. ZONE: Zeroth order nonconvex multi-agent optimization over networks. *IEEE Trans. Automat. Control*, 64:3995–4010, 2019.

[12] P.D. Hough, T. G. Kolda, and V. Torczon. Asynchronous parallel pattern search for nonlinear optimization. *SIAM J. Sci. Comput.*, 23:134–156, 2001.

[13] T. G. Kolda. Revisiting asynchronous parallel pattern search for nonlinear optimization. *SIAM J. Optim.*, 16:563–586, 2005.

[14] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Rev.*, 45:385–482, 2003.

[15] J. Larson, M. Menickelly, and S. M. Wild. Derivative-free optimization methods. *Acta Numer.*, 28:287–404, 2019.

[16] Z. Li and L. Chen. Communication-efficient decentralized zeroth-order method on heterogeneous data. In *2021 13th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6. IEEE, 2021.

[17] Y. Liu, T. Lin, A. Koloskova, and S. U. Stich. Decentralized gradient tracking with local steps. *Optim. Methods Softw.*, pages 1–28, 2024.

[18] J. J. Moré and S. M. Wild. Benchmarking derivative-free optimization algorithms. *SIAM J. Optim.*, 20:172–191, 2009.

[19] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Automat. Control*, 54(1):48, 2009.

[20] E. K. Ryu and W. Yin. *Large-Scale Convex Optimization: Algorithms & Analyses via Monotone Operators*. Cambridge University Press, 2022.

[21] A. K. Sahu, D. Jakovetic, D. Bajovic, and S. Kar. Distributed zeroth order optimization over random networks: A Kiefer-Wolfowitz stochastic approximation approach. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4951–4958. IEEE, 2018.

[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*, 108(11):1890–1905, 2020.

[23] S. M. Shah, Al. S. Berahas, and R. Bollapragada. Adaptive consensus: A network pruning approach for decentralized optimization. *SIAM J. Optim.*, 34(4):3653–3680, 2024.

[24] Y. Sun, G. Scutari, and D. Palomar. Distributed nonconvex multiagent optimization over time-varying networks. In *2016 50th Asilomar Conference on Signals, Systems and Computers*, pages 788–794. IEEE, 2016.

[25] Y. Tang, J. Zhang, and N. Li. Distributed zero-order algorithms for nonconvex multiagent optimization. *IEEE Trans. Control Netw. Syst.*, 8(1):269–281, 2020.

[26] L. N. Vicente. Worst case complexity of direct search. *EURO J. Comput. Optim.*, 1:143–153, 2013.

[27] K. Yuan, Q. Ling, and W. Yin. On the convergence of decentralized gradient descent. *SIAM J. Optim.*, 26(3):1835–1854, 2016.

[28] J. Zeng and W. Yin. On nonconvex decentralized gradient descent. *IEEE Trans. Signal Process.*, 66(11):2834–2848, 2018.