CYBERNETICS AND PHYSICS, VOL. 12, NO. 1, 2023, 16–22

PARTIALLY OBSERVED DISTRIBUTED OPTIMIZATION UNDER UNKNOWN–BUT–BOUNDED DISTURBANCES

Victoria Erofeeva
Laboratory "Control of Complex Systems"
Institute for Problems in Mechanical Engineering of RAS,
St. Petersburg State University
St. Petersburg, Russia
eva@ipme.ru

Natalia Kizhaeva
Laboratory for Analysis and Modeling of Social Processes
St. Petersburg State University
St. Petersburg, Russia
n.kizhaeva@spbu.ru

Article history:
Received 18.05.2023, Accepted 19.06.2023

Abstract

In this paper, we consider non-stationary distributed optimization with partially observed parameters with acceleration based on the estimate sequence proposed by Y. Nesterov. We formulate this partial observability as time-varying communication matrix defined for each parameter separately. We propose the new distributed algorithm combining the accelerated Simultaneous Perturbation Stochastic Approximation (SPSA) and the described communication scheme as well as show its theoretical properties. The simulation validates the proposed algorithm in multi-sensor multi-target tracking problem over delayed channels.

Key words

distributed optimization, networked systems, disturbances, communication constraints, computational constraints, estimation

1 Introduction

Nowadays, distributed networks emerge in many practical areas such as transportation, telecommunication, logistics, opinion dynamics, flocking behaviour, multi-vehicle networks [Yu et al., 2010], [Ren et al., 2007], [Granichin et al., 2012], etc. The problems arising in network control systems are the subject of ever-growing research interest. For example, large-scale systems may be influenced by communication bottlenecks. In that sense, it is reasonable to impose communication constraints. These constraints can be accounted for through sparsification techniques. In optimization, it gained special interest due to the need for communication-efficient distributed learning. In this field, the researchers proposed compression operators that produce sparse vectors to be sent over communication channels, see, e.g. [Horváth and Richtarik,] and references therein. On the other hand, sparse structure may reflect a property of a system itself. In multi-area state estimation, the vector to be optimized is divided into local and boundary subsets. The local variables should stay private while the boundary ones can be exchanged with neighboring computing nodes. Both approaches produce a decomposition of parameters and lead to partially observed distributed optimization. In general, distributed algorithms may be more efficient than centralized ones due to their resilience, local communication and processing. While centralized approaches may produce communication and computation bottlenecks in large-scale systems, distributed ones successfully overcome such difficulties.

Another important problem in optimization is the improvement of convergence rate. Acceleration techniques have been studied for several decades. Heavy ball, which is the gradient descent with momentum, is asymptotically optimal among gradient-based methods on quadratics [Polyak, 1964]. Its stochastic variant, where gradient is replaced by a stochastic estimator, is widely used in deep learning. Additionally, momentum and step-size parameters can be estimated without the knowledge regarding the Hessian's smallest singular value, in contrast to classical accelerated methods like Nesterov acceleration and Polyak momentum [Pedregosa and Scieur, 2020]. The heavy ball method with constant step-sizes has a long history. It is known, for example, to achieve optimal black-box worst-case complexity of quadratic convex optimization [Nemirovsky, 1992]. In [Nesterov and Spokoiny, 2017], the authors consider random derivative-free methods and provide them with some complexity bounds for different classes of convex optimization problems as well as accelerated methods for smooth convex derivative-free optimization. In [Vorontsova et al., 2019], the authors propose an accelerated gradient-free method with a non-Euclidean proximal operator. Paper [Gorbunov et al., 2022] describes an

accelerated method for smooth stochastic derivative-free optimization with two-point feedback. The latter paper considers additionally possibly adversarial noise in the objective function value and analyze how this noise affects the convergence rate of the estimates. Stochastic scenarios are typically challenging. In contrast to classical optimization problems, stochastic ones bring additional problems with convergence of the algorithms (for example, gradient descent or Nesterov's accelerated gradient) [Scieur, 2018]. Moreover, some accelerated methods like heavy ball or Nesterov's accelerated gradient may also exhibit nonmonotonic convergence due to peak effects [Polyak et al., 2018; Ahiyevich et al., 2018].

In this paper, we combine our research in accelerated techniques for non-stationary distributed optimization under unknown-but-bounded disturbances and partially observed reformulation for the consensus term. This paper continues the line of research devoted to the improved distributed Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm proposed in [Erofeeva and Granichin, 2023] and incorporates a new communication scheme.

The paper is organized as follows. The preliminary information is given in Section 2. A formal problem setting of non-stationary distributed optimization is given in Section 3. The partially observed communication matrix is described in Section 4. The main result is presented in Section 5. In Section 6, the efficiency of the proposed algorithm is illustrated through the numerical simulation. Section 7 concludes the paper.

2 Preliminaries

Let $(\Omega, \mathcal{F}, P)$ be the underlying probability space corresponding to sample space Ω, set of all events $\mathcal{F}$, and probability measure P. $\mathbb{E}$ denotes mathematical expectation. Let $\mathcal{F}_{t-1}$ be the σ-algebra of all probabilistic events which happened up to time instant $t = 1, 2, \ldots$, $\mathbb{E}_{\mathcal{F}_{t-1}}$ denotes the conditional mathematical expectation with respect to σ-algebra $\mathcal{F}_{t-1}$.

Throughout the paper, we represent d-dimensional column vectors as lowercase bold symbols (e.g. $\mathbf{x} = [x_1, \ldots, x_d]^{\mathrm{T}}$), and scalars as non-bold symbols. $\otimes$ is the Kronecker product. $[\cdot]^\top$ is the matrix or vector transpose. $|\cdot|$ is the cardinality of a set or the total number of unique elements in a set. $\mathbf{1}_d \in \mathbb{R}^d$ is a vector of all ones. $\mathrm{I}_d \in \mathbb{R}^{d\times d}$ is the identity matrix. $\mathbf{0}_d \in \mathbb{R}^d$ is a vector of all zeros.

3 Problem Statement

3.1 Networked System

Consider a networked system consisting of n nodes. Nodes are able to communicate with each other through a network described by undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, \ldots, n\}$ is a set of vertices and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is a set of edges. The vertices correspond to the areas while the edges represent the information flows between them. For node $i \in \mathcal{N}$, the set of *neighbors* is defined as $\mathcal{N}^i = \{j \in \mathcal{N} : (i, j) \in \mathcal{E}\}$. The *degree* of $i \in \mathcal{N}$ equals $|\mathcal{N}^i|$ and is defined as $\deg(i)$. A subgraph of $\mathcal{G}$ is a graph $\bar{\mathcal{G}} = (\mathcal{N}_{\bar{\mathcal{G}}}, \mathcal{E}_{\bar{\mathcal{G}}})$, where $\mathcal{N}_{\bar{\mathcal{G}}} \subseteq \mathcal{N}$ and $\mathcal{E}_{\bar{\mathcal{G}}} \subseteq \mathcal{E}$.

Alternatively, we express the communication between nodes in a matrix form through a communication matrix for which we adopt the following definition from [Makhdoumi and Ozdaglar, 2017]:

Definition 1 (Communication matrix). *Let $\mathcal{W}$ be a $m \times m$ matrix whose entries satisfy the following property. For any $i \in \mathcal{N}$, $\mathcal{W}_{ij} = 0$ for $j \notin \mathcal{N}^i$. We refer to $\mathcal{W}$ as the communication matrix.*

Next, we impose the assumptions on the communication matrix and graph $\mathcal{G}$.

Assumption 1. *The communication matrix $\mathcal{W}$ satisfies* $\mathrm{null}(\mathcal{W}) = \mathrm{span}(\mathbf{1}_n)$, *where* $\mathrm{null}(\mathcal{W})$ *denotes the nullspace of $\mathcal{W}$.*

Assumption 2. *Graph $\mathcal{G}$ is connected, i.e., there is a path between every pair of distinct vertices of $\mathcal{G}$.*

Taking into account Assumptions 1 and 2, one of the available choices for communication matrix is the Laplacian of $\mathcal{G}$, which is $\mathcal{L}(\mathcal{G}) = (l_{ij})_{i,j\in\mathcal{N}}$:

$$l_{ij} = \begin{cases} -1 \ \text{ if } \ (i,j) \in \mathcal{E}, \\ \deg(i) \ \text{ if } \ i = j, \\ 0 \ \text{ otherwise.} \end{cases}$$

3.2 Distributed Non-stationary Mean-risk Optimization

Let $\theta_t = [\theta_t^1, \ldots, \theta_t^m]^{\mathrm{T}} \in \mathbb{R}^{md}$ be a vector of unknown parameters $\theta_t^j \in \mathbb{R}^d, j = 1, \ldots, m$ to be estimated at time instant $t = 1, 2, \ldots$. Each parameter evolves in accordance with a state-transition model:

$$\theta_t^j = A_t^j \theta_{t-1}^j + \xi_t^j, \tag{1}$$

where $A_t^j \in \mathbb{R}^{md\times md}$ is a transition matrix, $\{\xi_t^j\}$, $\xi_t \in \Xi$, is a non-controllable deterministic (e.g., $\Xi = \mathbb{N}$ and $\xi_t = t$) or random sequence. In the latter case we assume that a probability distribution of ξ_t exists and may be known or unknown. This sequence indicates abrupt changes in the dynamics (e.g., in target tracking problems, maneuvers of moving objects, or, in power systems, sudden changes in system operating conditions due to power injections), a disturbance (e.g., discretization and model approximation errors, external signal injection during a cyber-attack).

Remark: model (1) is widely used in the works devoted to state estimation, e.g., power system dynamic state estimation [Zhao et al., 2019].

Each node collects measurements represented by a linear model:

$$\mathbf{z}_t^i = H^i \theta_t + \mathbf{w}_t^i, \tag{2}$$

where $H^i \in \mathbb{R}^{l\times md}$ is a measurement matrix, $\mathbf{w}_t^i \in \mathbb{R}^l$ is the noise following Gaussian distribution with zero mean and standard deviation σ_w. Then, the problem is to find estimate $\widehat{\theta}_t$ of unknown parameter θ_t minimizing function

$$f_{\xi_t}(\widehat{\theta}_t, \mathbf{z}_t) = ||H\bar{\theta}_t - \mathbf{z}_t||^2. \tag{3}$$

based on aggregated measurements $\mathbf{z}_t^i$ received by a fusion center. Here, $H \in \mathbb{R}^{nl\times nmd}$ is a block-diagonal matrix consisting of H^i on its diagonal, $\bar{\theta}_t = \mathbf{1}_n \otimes \widehat{\theta}_t$, $\mathbf{z}_t = [\mathbf{z}_t^1, \dots, \mathbf{z}_t^n]^{\mathrm{T}}$.

In many applications, first-order methods have the bottleneck appearing due to the computation of ∇f. To motivate this statement, let us mention a few such examples:

> Evaluating the gradient of (3) requires $\mathcal{O}(n^2mdl)$ arithmetic operations. In large-scale applications, it becomes unrealistic to calculate the gradient in real-time.
>
> In some cases, the computation of ∇f involves a black-box simulation procedure. Then, it's impossible to obtain the gradient implicitly.

This work considers zeroth-order optimization, where we have only measurements of function to be optimized. We obtain ∇f through stochastic approximation via finite differences instead of implicit calculations [Kiefer et al., 1952]. In finite differences, we observe how the function behaves around a current point. For this purpose, we introduce a sequence of controllable measurement points $\mathbf{x}_1, \mathbf{x}_2, \dots$ chosen according to an observation plan, e.g., $\mathbf{x}_t = \widehat{\theta}_t \pm \epsilon$, ϵ is a random variable drawn from a known distribution.

Remark: In the example of observation plan, we use $\widehat{\theta}_t$ with the assumption that θ_t is observable. In some applications, we cannot directly measure value of θ_t. However, it is possible to make an observation plan consisting of quantities that influence the estimating parameter and we can indirectly observe how it evolves. One such example could be found in [Amelina et al., 2015].

The values $y_1, y_2, \dots$ of the functions $f_{\xi_t}(\cdot)$ are observable at every time instant t with additive external *unknown-but-bounded* noise v_t

$$y_t = f_{\xi_t}(\mathbf{x}_t, \mathbf{z}_t) + v_t. \tag{4}$$

Equation (4) is called a stochastic zeroth-order oracle that returns a noisy value of function $f_{\xi_t}(\cdot)$.

Problem (3) requires a centralized optimization procedure. In practice, centralized framework is subject to performance limitations, such as a single point of failure, high communication requirement, and substantial computation burden. All of these aspects have influenced the development of distributed approaches. Thereby, we consider an optimization problem in which the cost function $\bar{F}_t(\widehat{\theta}_t, \mathbf{z}_t)$ is expressed as the sum of local contributions $F_t^i(\widehat{\theta}_t^i, \mathbf{z}_t^i) = \mathbb{E}_{\mathcal{F}_{t-1}} f_{\xi_t}^i(\widehat{\theta}_t^i, \mathbf{z}_t^i)$ and all of them have a common minimizer. Moreover, minimizer $\widehat{\theta}_t^\star$ of $\bar{F}_t(\widehat{\theta}_t, \mathbf{z}_t)$ may vary over time. Formally, the *non-stationary mean-risk optimization problem* is as follows: estimate the time-varying minimum point $\widehat{\theta}_t^\star$ of the distributed function

$$\begin{aligned}\bar{F}_t(\widehat{\theta}_t, \mathbf{z}_t) &= \sum_{i\in\mathcal{N}} F_t^i(\widehat{\theta}_t^i, \mathbf{z}_t^i) \\ &= \mathbb{E}_{\mathcal{F}_{t-1}} \sum_{i\in\mathcal{N}} f_{\xi_t}^i(\widehat{\theta}_t^i, \mathbf{z}_t^i) \to \min_{\widehat{\theta}_t}.\end{aligned} \tag{5}$$

In the distributed setting, the sensors construct sequence of measurement points $\mathbf{x}_1^i, \mathbf{x}_2^i, \dots$ and collect $y_1^i, y_2^i, \dots$ independently from each other based on their own measurements $\mathbf{z}_t^i$ and current estimate $\widehat{\theta}_t^i$.

3.3 Partially Observed Optimization: Motivation

In wide-area estimation problems, the underlying interconnected system is usually partitioned into regions based on some criteria, e.g., range of sensors. Such problems occur, for example, in multi-target tracking. The system states of each region are monitored and managed by a local control unit referred to as node. The overall goal is to estimate the states in each region in an optimal way. In general, the nodes can estimate their local states without communicating with neighboring regions. However, it may bring some issues: estimated solution may be sub-optimal; the estimates obtained by different nodes for the same target should be consistent, then there is a need for communication between those regions in order to utilize the measurements.

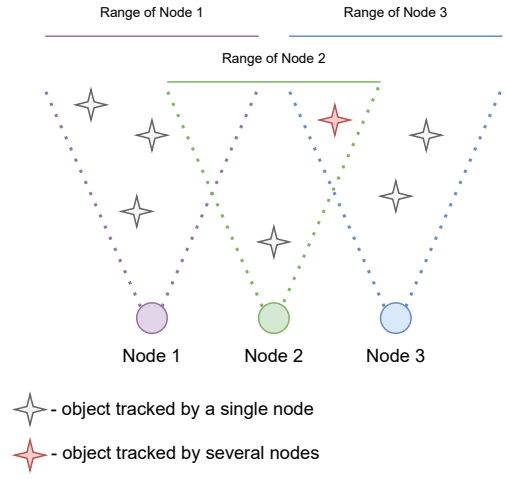

Figure 1. Schematic representation of a system

For illustration purposes, consider Figure 1. There are three regions and nodes corresponding to them. Each node estimates parameters of the targets detected in their range. The ranges intersect and some targets are detected by several nodes simultaneously. The parameters

of these targets form boundary variables for neighboring nodes. Therefore, towards optimal estimation, the regions should communicate with each other. This could be done through information sharing of states related to boundary variables. Hence, for a partially-observed state estimation, each region would align shared states with neighboring regions when performing local estimation.

4 Partially Observed Parameters

The described formulation is a common one for distributed unconstrained optimization. However, in practice, we frequently have a decomposition of optimization variable into private and boundary subsets. The private part is solely optimized by the node owning this information and corresponding measurements. We express this part as possibly sparse vector $\widehat{\theta}_t^i$. The boundary part can be optimized by several nodes and requires the information exchange with neighboring nodes. Thus, we transform the initial distributed problem into partially observed distributed problem. The process is schematically represented on Figure 2 and described in subsequent steps.

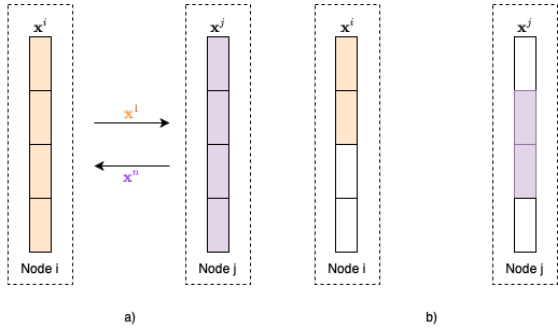

Figure 2. Decomposition and variable exchange. a) Conventional distributed setting; b) Partially-observed optimization.

4.1 Decomposition

Figure 2a illustrates the common scheme of distributed optimization between two arbitrary nodes i and j. Both nodes optimize the whole vector and send it to the neighboring node. Next, we move to Figure 2b. Let $\mathcal{X} = \{1, \dots, d\}$ be a set of indices corresponding to the entries contained in $\widehat{\theta}_t^i$. We divide this set into n subsets $\mathcal{X}_t^1, \dots, \mathcal{X}_t^n$ such as

$$\mathcal{X}_t^i \cap \mathcal{X}_t^j = \mathcal{X}_t^{i,j}, \;\; |\mathcal{X}_t^{i,j}| \in \{0, \dots, d\}. \tag{6}$$

The latter means that the intersection between two arbitrary sets i and j is set $\mathcal{X}_t^{i,j}$, which has a size ranging from 0 to d elements.

Consider arbitrarily chosen set of indices $\mathcal{S}_t \subseteq \mathcal{X}$ and vector $\mathbf{s} \in \mathbb{R}^d$. We denote by $B_{\mathcal{S}_t} = [\mathbf{e}_{\omega_1}^{\mathrm{T}}, \dots, \mathbf{e}_{\omega_{|\mathcal{S}_t|}}^{\mathrm{T}}]$ the selection matrix. Here and after, $\mathbf{e}_l \in \mathbb{R}^d$ is the canonical basis vector that has a unit entry at the selected index l and zeros elsewhere and $\omega_l \in \mathcal{S}_t$. Then, we define a linear map:

$$\Gamma(B_{\mathcal{S}_t}, \mathbf{s}) = B_{\mathcal{S}_t}^{\mathrm{T}} B_{\mathcal{S}_t} \mathbf{s}. \tag{7}$$

This linear map produces a sub-vector of $\mathbf{s}$ taking the entries which indices are contained in $\mathcal{S}$. Then, it takes this sub-vector and restores initial vector dimension filling the rest of the entries by zeros.

Finally, we get a vector to be optimized by node i:

$$\widehat{\theta}_t^i = \Gamma\left(B_{\mathcal{X}_t^i}, \theta_t\right). \tag{8}$$

4.2 Communication Matrix

Communication matrix for the partially observed parameters can be adopted from [Erofeeva et al., 2023]:

$$\mathcal{W}_t = \sum_{l \in \mathcal{X}^b} \mathcal{L}(\bar{\mathcal{G}}_t^l) \otimes \mathbf{e}_l \mathbf{e}_l^{\top}, \tag{9}$$

where $\mathcal{X}^b = \bigcup_{i \in \mathcal{N}, j \in \mathcal{N}^i} \mathcal{X}_t^{i,j}$, $\mathbf{e}_l \in \mathbb{R}^d$ is the canonical basis vector that has a unit entry at the selected index l and zeros elsewhere, $\bar{\mathcal{G}}_t^l$ is a subgraph of $\mathcal{G}$ at time instant t associated with the parameter at index l.

The components of canonical vector can be formed in different ways. When communication constraints should be accounted for, we can artificially choose these components in deterministic or random manner. Also, these components may reflect the structure of the problem itself and appear naturally due to this factor (e.g., multi-area state estimation, target tracking with limited sensor range, etc.). The next subsection describes the randomized way of generating these components.

4.2.1 Random Components Our previous works rely on a random sparsification strategy used to satisfy communication and sensing constraints. Here, we utilize the concept of compression operator, which includes sparsification as a special case and generalizes our previous approach.

Definition [Islamov et al., 2021]. A possibly randomized map $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ is a compression operator if there exists a constant $\omega \geq 0$ such that the following relations hold for all $\mathbf{x} \in \mathbb{R}^d$:

$$\mathbb{E}[\mathcal{C}(\mathbf{x})] = \mathbf{x} \quad \text{(unbiasedness)} \tag{10}$$

$$\mathbb{E}[\|\mathcal{C}(\mathbf{x})\|^2] \leq (1 - \omega)\|\mathbf{x}\|^2 \quad \text{(variance bound)} \tag{11}$$

In particular, the compression operator has the form defined below. We use the similar strategy as before, but enhance our theoretical analysis through generalization based on the definition of compression operator.

Definition [Stich et al., 2018]. For a parameter $1 \leq p \leq d$, the compression operator $\mathcal{C}_p : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is defined as

$$\mathcal{C}_p(\mathbf{x}, \pi) = \frac{d}{p} \mathrm{diag}_d(\pi)\mathbf{x}, \tag{12}$$

where $\pi \in \mathbb{R}^d$ is a random vector uniformly distributed on discrete set $\Omega_d = \binom{[d]}{p}$, which denotes the set of all p element subsets of $[d]$. The variance parameter associated with this operator is $\omega = 1 - \frac{d}{p}$. We abbreviate $\mathcal{C}_p(\mathbf{x})$ whenever the second argument is chosen uniformly at random.

Then, the canonical vectors and the corresponding subgraphs can be generated by applying compression operator $\mathcal{C}_p(\mathbf{x})$ to the vector to be optimized/estimated.

The next section combines the new formulation of communication matrix and accelerated distributed SPSA-based consensus algorithm.

5 Main Result

This section introduces a new algorithm with partially observed parameters starting with the distributed SPSA-based consensus algorithm followed by its accelerated version. After that, we end this section providing the convergence analysis.

5.1 Recap of SPSA-based consensus algorithm (DSPSA)

SPSA-based consensus algorithm is the Simultaneous Perturbation Stochastic Approximation equipped with a consensus-based procedure. In sequel, we describe its main components and steps.

Let $\mathbf{\Delta}_k^i,\ k = 1, 2, \ldots,\ i \in \mathcal{N}$, be an observed sequence of independent random vectors in $\mathbb{R}^d$, called the *simultaneous test perturbation*, drawn from Bernoulli distribution. Each component of the vector independently takes value $\pm\frac{1}{\sqrt{d}}$ with probability $\frac{1}{2}$.

Let us take fixed nonrandom initial vectors $\widehat{\theta}_0^i \in \mathbb{R}^d$, positive step-size h, gain coefficient ω, and parameter $\beta > 0$. We consider the algorithm with two observations of distributed sub-functions $f_{\xi_t}^i(\theta)$ for each agent $i \in \mathcal{N}$ for constructing sequences of measurement points $\{\mathbf{x}_t^i\}$ and estimates $\{\widehat{\theta}_t^i\}$:

$$\begin{cases} \mathbf{x}_{2k}^i = \widehat{\theta}_{2k-2}^i + \beta\mathbf{\Delta}_k^i,\ \mathbf{x}_{2k-1}^i = \widehat{\theta}_{2k-2}^i - \beta\mathbf{\Delta}_k^i, \\ \widehat{\theta}_{2k-1}^i = \widehat{\theta}_{2k-2}^i, \\ \widehat{\theta}_{2k}^i = \widehat{\theta}_{2k-1}^i - h\left(\frac{y_{2k}^i - y_{2k-1}^i}{2\beta}\mathbf{\Delta}_k^i + \right. \\ \qquad\qquad \left. \omega\sum_{j\in\mathcal{N}^i} b^{i,j}(\widehat{\theta}_{2k-1}^i - \widehat{\theta}_{2k-1}^j)\right). \end{cases} \tag{13}$$

5.2 Accelerated DSPSA with Partially Observed Parameters

We modify the accelerated version of DSPSA presented in [Erofeeva and Granichin, 2023] by equipping it with the partially observed communication matrix.

Taking into account the assumptions from [Erofeeva and Granichin, 2023], where L is Lipschitz constant and μ is the constant related to strong convexity, we define a list of variables. At each node, we choose initial estimate $\hat{\theta}_0^i \in \mathbb{R}^d$, and parameters $\gamma_0^i > 0$, $h > 0$, $\beta > 0$, $\eta \in (0, \mu)$, $\alpha_0^i \in (0, 1)$. We also define $z_0^i = \hat{\theta}_0^i$ and $H = h - \frac{h^2 L}{2}$ and pick $\alpha_x^i \in (0, 1)$. At each $k > 0$, we find α_k by solving the equation presented in the paper mentioned above as well as γ_k^i.

We present an algorithm that requires two measurements of function $f_{\xi_t}^i(\cdot)$ taken subsequently. Using gradient approximation techniques, we produce estimates $\{\widehat{\theta}_t^i\}$ at $k \geq 1$ and each node:

$$\begin{cases} \tilde{\mathbf{x}}_{2k-2}^i = \frac{1}{\gamma_{k-1}^i + \alpha_k^i(\mu-\eta)}\left(\alpha_k^i\gamma_{k-1}^i\mathbf{z}_{2k-2}^i + \gamma_k^i\widehat{\theta}_{2k-2}^i\right), \\ \mathbf{x}_{2k}^i = \tilde{\mathbf{x}}_{2k-2}^i + \beta\mathbf{\Delta}_k^i,\ \mathbf{x}_{2k-1}^i = \tilde{\mathbf{x}}_{2k-2}^i - \beta\mathbf{\Delta}_k^i, \\ \tilde{\mathbf{x}}_{2k-1}^i = \tilde{\mathbf{x}}_{2k-2}^i,\ \widehat{\theta}_{2k-1}^i = \widehat{\theta}_{2k-2}^i, \\ \mathbf{g}_{2k}^i = \mathbf{\Delta}_k^i\frac{y_{2k}^i - y_{2k-1}^i}{2\beta} + \\ \omega\sum_{j\in\mathcal{N}^i}\sum_{l\in\mathcal{X}_t^{i,j}}\mathbf{e}_l \otimes b_{k,l}^{i,j}(\tilde{\mathbf{x}}_{2k-1,l}^i - \tilde{\mathbf{x}}_{2k-1}^{i,j,l}), \\ \widehat{\theta}_{2k}^i = \tilde{\mathbf{x}}_{2k-1}^i - h\mathbf{g}_{2k}^i, \\ \mathbf{z}_{2k}^i = \frac{1}{\gamma_k^i}\Big[(1-\alpha_k^i)\gamma_{k-1}^i\mathbf{z}_{2k-2}^i + \\ \alpha_k^i(\mu-\eta)\tilde{\mathbf{x}}_{2k-1}^i - \alpha_k^i\mathbf{g}_{2k}^i)\Big], \end{cases}$$

where B_k is an adjacency matrix corresponding to $\mathcal{W}_k$.

5.3 Convergence Analysis

Proposition 1: Let $\bar{\lambda}_m$ be defined as

$$\bar{\lambda}_m = \max_t \max_{1\leq l\leq d} \lambda_{\max}^{\frac{1}{2}}(\mathcal{L}(\bar{\mathcal{G}}_t^l)^{\mathrm{T}}\mathcal{L}(\bar{\mathcal{G}}_t^l)),$$

then we obtain the convergence properties similar to Theorem 1 in [Erofeeva and Granichin, 2023].

Proof: The convergence of the proposed algorithm depends on the spectral properties of the underlying communication matrix. We can obtain the result of Theorem 1 from [Erofeeva and Granichin, 2023] by redefining the constant related to the graph properties. Hence, let us analyze $\bar{\lambda}_m = \lambda_{\max}^{\frac{1}{2}}(\mathcal{W}^{\mathrm{T}}\mathcal{W})$.

Based on Lemma 2 in [Chezhegov et al., 2022], it follows that

$$\lambda_{\max}^{\frac{1}{2}}(\mathcal{W}^{\mathrm{T}}\mathcal{W}) = \max_t \max_{1\leq l\leq d} \lambda_{\max}^{\frac{1}{2}}(\mathcal{L}(\bar{\mathcal{G}}_t^l)^{\mathrm{T}}\mathcal{L}(\bar{\mathcal{G}}_t^l)).$$

Substituting the redefined constant, we get the result of Theorem 1 in [Erofeeva and Granichin, 2023].

6 Simulations

We consider multi-sensor multi-target tracking problem fully described in [Granichin et al., 2020]. The sensor network consisting on n nodes spatially distributed over an area of interest tracks m targets. The sensor nodes are assumed to be static. Their state is represented by a position in 2D plane. The state-transition model of targets is defined as in (1). The sensors can measure the distance between their positions and the positions of targets. The goal of the sensor network is to estimate the unknown target positions based on the measured distances. The sensors are able to communicate with each

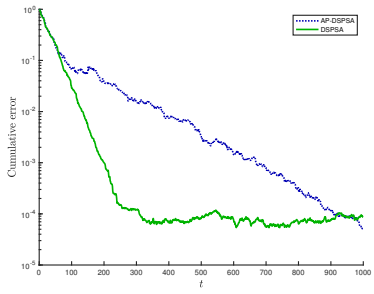

Figure 5. Max delay in AP-DSPSA equals 150 iterations. SPSA doesn't have any delays.

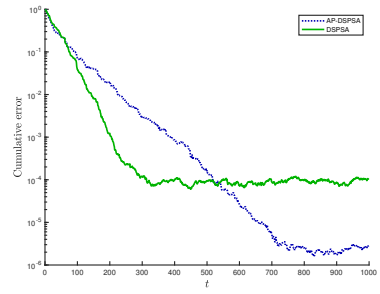

Figure 4. Max delay in AP-DSPSA equals 50 iterations. SPSA doesn't have any delays.

other over possibly delayed channels to obtain a common solution.

We estimate the target positions using the proposed accelerated SPSA algorithm with partially observed parameters. We have set the following parameters of the algorithm: $h = 0.08, \omega = 1, \beta = 0.5, \eta = 0.95, \alpha_x^i = 0.1, \gamma_0^i = 2$, $L = 2, \mu = 2$. The initial estimates $\widehat{\theta}_0^i$ are chosen randomly at each sensor. The states of the sensors are chosen randomly from interval $[100; 120]$. The number of sensors is $n = 3$, the number of targets is $m = 10$.

In the simulation, the new algorithm AP-DSPSA is compared with the previous one from [Granichin et al., 2020]. Figures 3, 4, 5 show the cumulative tracking error in different delayed scenarios on the logarithmic scale. The delays are added to the new algorithm only. We generate them randomly setting the max delay equal to 10, 50 and 150 iterations, correspondingly. As can be seen, the delays influence the convergence time. However, the algorithm is still able to achieve the accuracy level of the non-delayed version.

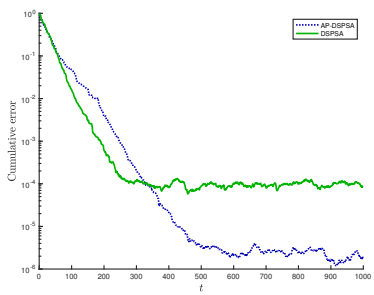

Figure 3. Max delay in AP-DSPSA equals 10 iterations. SPSA doesn't have any delays.

7 Conclusion

In this paper, we consider non-stationary distributed optimization with partially observed parameters. We formulate this partial observability as time-varying communication matrix defined for each parameter separately. We propose the new AP-DSPSA algorithm combining the accelerated SPSA and the described communication scheme as well as show its theoretical properties. The simulation validates the proposed algorithm in target tracking problem over delayed channels.

Acknowledgements

This work was mainly supported (Sections 1–5) by Russian Science Foundation (project no. 22-71-00072, `https://rscf.ru/en/project/22-71-00072/`). Experimental results in Section 6 were supported by the St. Petersburg State University (project ID 94062114).

References

Ahiyevich, U., Parsegov, S. E., and Shcherbakov, P. S. (2018). Upper bounds on peaks in discrete-time linear systems. *Automation and Remote Control*, **79**, pp. 1976–1988.

Amelina, N., Erofeeva, V., Granichin, O., and Malkovskii, N. (2015). Simultaneous perturbation stochastic approximation in decentralized load balancing problem. *IFAC-PapersOnLine*, **48** (11), pp. 936–941.

Chezhegov, S., Novitskii, A., Rogozin, A., Parsegov, S., Dvurechensky, P., and Gasnikov, A. (2022). A general framework for distributed partitioned optimization. *IFAC-PapersOnLine*, **55** (13), pp. 139–144.

Erofeeva, V. and Granichin, O. (2023). Improved simultaneous perturbation stochastic approximation-based consensus algorithm for tracking. In *31st Mediterranean Conference on Control and Automation (MED)*.

Erofeeva, V., Parsegov, S., Osinenko, P., and Kamal, S. (2023). Distributed state estimation for multi-area data reconciliation. In *31st Mediterranean Conference on Control and Automation (MED)*.

Gorbunov, E., Dvurechensky, P., and Gasnikov, A. (2022). An accelerated method for derivative-free smooth stochastic convex optimization. *SIAM Journal on Optimization*, **32** (2), pp. 1210–1238.

Granichin, O., Erofeeva, V., Ivanskiy, Y., and Jiang, Y. (2020). Simultaneous perturbation stochastic approximation-based consensus for tracking under unknown-but-bounded disturbances. *IEEE Transactions on Automatic Control*, **66** (8), pp. 3710–3717.

Granichin, O., Skobelev, P., Lada, A., Mayorov, I., and Tsarev, A. (2012). Comparing adaptive and non-adaptive models of cargo transportation in multi-agent system for real time truck scheduling. *Proceedings of the 4th International Joint Conference on Computational Intelligence*, pp. 282–285.

Horváth, S. and Richtarik, P. A better alternative to error feedback for communication-efficient distributed learning. In *International Conference on Learning Representations*.

Islamov, R., Qian, X., and Richtárik, P. (2021). Distributed second order methods with fast rates and compressed communication.

Kiefer, J., Wolfowitz, J., et al. (1952). Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, **23** (3), pp. 462–466.

Makhdoumi, A. and Ozdaglar, A. (2017). Convergence rate of distributed admm over networks. *IEEE Transactions on Automatic Control*, **62** (10), pp. 5082–5095.

Nemirovsky, A. S. (1992). Information-based complexity of linear operator equations. *Journal of Complexity*, **8** (2), pp. 153–175.

Nesterov, Y. and Spokoiny, V. (2017). Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, **17** (2), pp. 527–566.

Pedregosa, F. and Scieur, D. (2020). Acceleration through spectral density estimation. In *International Conference on Machine Learning*, PMLR, pp. 7553–7562.

Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, **4** (5), pp. 1–17.

Polyak, B. T., Shcherbakov, P. S., and Smirnov, G. (2018). Peak effects in stable linear difference equations. *Journal of Difference Equations and Applications*, **24** (9), pp. 1488–1502.

Ren, W., Beard, R., and Atkins, E. (2007). Information consensus in multivehicle cooperative control. *Control Systems, IEEE*, **27** (2), pp. 71–82.

Scieur, D. (2018). *Acceleration in optimization*. PhD thesis, Université Paris sciences et lettres.

Stich, S. U., Cordonnier, J.-B., and Jaggi, M. (2018). Sparsified sgd with memory. *Advances in Neural Information Processing Systems*, **31**, pp. 4447–4458.

Vorontsova, E. A., Gasnikov, A. V., Gorbunov, E. A., and Dvurechenskii, P. E. (2019). Accelerated gradient-free optimization methods with a non-euclidean proximal operator. *Automation and Remote Control*, **80** (8), pp. 1487–1501.

Yu, W., Chen, G., and Cao, M. (2010). Distributed leader–follower flocking control for multi-agent dynamical systems with time-varying velocities. *Systems & Control Letters*, **59** (9), pp. 543–552.

Zhao, J., Gómez-Expósito, A., Netto, M., Mili, L., Abur, A., Terzija, V., Kamwa, I., Pal, B., Singh, A. K., Qi, J., et al. (2019). Power system dynamic state estimation: Motivations, definitions, methodologies, and future work. *IEEE Transactions on Power Systems*, **34** (4), pp. 3188–3198.