Distributed Momentum Based Multi-Agent Optimization with Different Constraint Sets

Xu Zhou, Zhongjing Ma*, Suli Zou*, and Kostas Margellos

***Abstract*— This paper considers a class of consensus optimization problems over a time-varying communication network wherein each agent can only interact with its neighbours. The target is to minimize the summation of all local and possibly non-smooth objectives in the presence of different constraint sets per agent. To achieve this goal, we propose a novel distributed heavy-ball algorithm that combines the subgradient tracking technique with a momentum term related to history information. This algorithm promotes the distributed application of existing centralized accelerated momentum methods, especially for constrained non-smooth problems. Under certain assumptions and conditions on the step-size and momentum coefficient, the convergence and optimality of the proposed algorithm can be guaranteed through a rigorous theoretical analysis, and a convergence rate of $\mathcal{O}(\ln k/\sqrt{k})$ in objective value is also established. Simulations on an ℓ_1-regularized logistic-regression problem show that the proposed algorithm can achieve faster convergence than existing related distributed algorithms, while a case study involving a building energy management problem further demonstrates its efficacy.**

***Index Terms*— Distributed optimization, multi-agent networks, heavy-ball momentum, sub-gradient averaging consensus.**

I. Introduction

GRADIENT descent algorithms have been extensively employed in the distributed optimization and machine learning [1], [2] literature, with numerous applications in different complex systems such as wireless networks [3], robotics [4] and multi-energy systems [5]. To accelerate the convergence of gradient descent methods, a momentum term involving the information of previous iterates has been introduced in the literature. The concept of momentum comes from physics which describes a moving object that still keeps moving without the intervention of out-side forces. The algorithms with momentum could move more quickly when the plateaus in the error surface exist. One of the typical momentum methods is heavy-ball momentum which has been extensively used to train deep network models and has made remarkable progress in various applications [6]. When the objective function is twice continuously differentiable and strongly convex, the heavy-ball method has a linear convergence rate and a better convergence factor than both gradient descent and Nesterov's accelerated gradient methods [7]. However, the theoretical analysis about the optimality and convergence of heavy-ball methods is still challenging, especially for non-smooth convex problems. Although [8] and recently [9] provide a convergence analysis of the heavy-ball method for non-smooth problems, they are both centralized and do not involve local objectives.

Only a few papers have studied the distributed heavy-ball optimization, and most of these papers such as [10]–[14] require smoothness and strong-convexity of the objective functions. Specifically, the authors in [10] proposed a distributed heavy-ball method, denoted as the $\mathcal{AB}m$ algorithm to solve an optimization problem which aims to minimize the summation of local objectives in a multi-agent setting with gradient tracking. The algorithm had a global R-linear rate under certain conditions. Papers [11] and [14] developed the Nesterov's gradient and heavy-ball double accelerated distributed algorithms for strongly convex objective functions, which could achieve linear convergence. A family of parametric distributed momentum methods was proposed in [12] for the smooth and strongly convex functions, which included the results in [10]. With different choices of the momentum parameter, it could obtain different distributed momentum methods. Reference [13] extended the $\mathcal{AB}m$ algorithm proposed in [10] to time-varying directed networks leveraging a gradient-tracking technique with linear convergence still being achieved.

However, these distributed heavy-ball methods only study unconstrained optimization problems, i.e., $\min_{x\in\mathbb{R}^n} f(x) = \sum_{i=1}^N f_i(x)$, where f_i denotes the objective function of agent i, $i = 1, \ldots, N$. They could not be directly applied to solve the constrained case, since simply introducing constraints into the objective such as via an indicator function could lead to non-smoothness and violate the boundedness assumption of subgradients. So far, there are only few results on distributed heavy-ball for non-smooth constrained problems. Hence, this paper is motivated to fill this gap and exploit the potential strength of heavy-ball momentum for accelerated convergence. More specifically, we focus on a general class of convex optimization problems with a separable objective function where

This work was supported by the National Natural Science Foundation (NNSF) of China under Grant 62003037. Research was performed in part when the first author was visiting the University of Oxford. For the purpose of Open Access, K. Margellos has applied a CC BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.

* Corresponding authors: Zhongjing Ma and Suli Zou.

Xu Zhou, Zhongjing Ma and Suli Zou are with the school of Automation, Beijing Institute of Technology (BIT), and the National Key Laboratory of Complex System Intelligent Control and Decision (BIT), Beijing 100081, China. Emails: {zhouxu0879, mazhongjing, sulizou}@bit.edu.cn. Kostas Margellos is with the Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, U.K. Email: kostas.margellos@eng.ox.ac.uk.

TABLE I: Comparison between the proposed method and related distributed methods. Legend: Strongly stands for strongly convex objective functions.

	Convex	Non-smooth	Heterogeneous constraints	Time-varying	(Sub)-gradient averaging	Momentum acceleratiion
[10]–[12]	Strongly	×	×	×	✓	✓
[13], [14]	Strongly	×	×	✓	✓	✓
[15]	✓	✓	×	✓	×	×
[16], [17]	✓	✓	×	×	✓	×
[18]	✓	×	✓	✓	×	×
[19]	✓	×	✓	×	×	×
[20], [21]	✓	✓	✓	×	✓	×
[22], [23]	Strongly	✓	✓	✓	×	×
[24]–[27]	✓	✓	✓	✓	×	×
[28]	✓	✓	✓	✓	✓	×
Our work	✓	✓	✓	✓	✓	✓

all agents have a common decision vector, considering heterogeneous constraints as well as a time-varying communication network, allowing for a broader class of applications. The problem is formulated as $\min_{x\in\cap_{i=1}^N \mathcal{X}_i} f(x) = \sum_{i=1}^N f_i(x)$, where $\mathcal{X}_i$ denotes the constraint set of agent $i = 1, \cdots, N$. This paper aims at solving this class of problems by applying a heavy-ball momentum algorithm which performs better than the ones proposed in the literature.

For the aforementioned constrained and possibly non-smooth convex problems, existing results on distributed optimization such as [15]–[17], [29], [30] considered cases where all agents had the same local constraints. Under heterogeneous constraints, the converged fixed points obtained via the methods in [15]–[17], [29], [30] might be local optimal solutions for the local objective functions, rather than the global consensus solution. The distributed algorithms proposed in [18], [19] could handle heterogeneous constraints, but they required differentiability of the objective functions. Reference [24] is the first distributed work considering heterogeneous constraints and without assuming differentiability of local objective functions. There have also been other distributed methods such as [20], [21] which can deal with this set-up. Reference [21] introduced a proximal-tracking distributed algorithm for this non-smooth problem with heterogeneous constraints by integrating the dynamic average consensus and adopting a constant step-size, which exhibited faster convergence than the P-EXTRA algorithm in [31]. Differently from [21], the methods proposed in [20] only require a row stochastic weight matrix, and the analysis of convergence rate considers a diminishing step-size. However, the algorithms designed in [20], [21] were implemented under a time-invariant network.

Only a few papers solve optimization problems like the one proposed in this paper over a time-varying network. Specifically, [25] proposed a distributed projected subgradient method to minimize the local objective function where the decisions of each agent are subject to different convex sets. Reference [26] extended the algorithm in [25] to the case of switched graphs and communication delays. In [22], a push-sum based constrained optimization algorithm was designed and a convergence rate of $\mathcal{O}(\frac{\ln k}{\sqrt{k}})$ was achieved over time-varying unbalanced directed topologies under the condition that at least one local objective function was strongly convex, which was different from the Fenchel dual gradient methods in [23] that required strong convexity of all local objective functions. Reference [27] designed a distributed scheme based on a proximal minimization perspective which could handle different sets of uncertainty scenarios. In [28], a distributed algorithm was developed by applying the same subgradient averaging technique as [16], [20], which could achieve a better convergence rate than [25], [27]. Our work extends the result in [28] by allowing the subgradients of all agents to be calculated at their own local estimates rather than the weighted estimates related to their neighbours' information, which enables less information transfer and fewer communication rounds. In addition, our designed algorithm has a faster convergence than the one in [28] by introducing an accelerated momentum term. For a quick overview, we compare our algorithm with the most closely related distributed methods in Table I.

The main contributions of this paper are summarized as follows:

1) To the best of our knowledge, the proposed heavy-ball algorithm is the first distributed algorithm that applies the heavy-ball momentum accelerated paradigm to solve non-smooth convex optimization problems subject to heterogeneous constraint sets per agent. In addition, the scheme can also handle time-varying communication networks.
2) A convergence rate of $\mathcal{O}(\frac{\ln k}{\sqrt{k}})$ in objective value can be obtained by applying the proposed distributed algorithm that involves the dynamic average consensus and accelerated momentum term. A rigorous analysis of the convergence and optimality of the algorithm is provided. Even though several lemmas are motivated by [27], [28], our main theoretical analysis is substantially different, constituting a nontrivial extension.
3) The simulation results on ℓ_1-regularized logistic regression verify the efficacy of the proposed heavy-ball algorithm with appropriate momentum parameter values, and demonstrate that the algorithm achieves a faster convergence compared to other main distributed methods, especially over sparse communication network. Moreover, we illustrate our algorithm on a case study involving a building energy management problem.

The rest of the paper is structured as follows. In Section II,

we formalize the constrained optimization problem and introduce the heavy-ball method. In Section III, we propose the accelerated distributed heavy-ball algorithm and relate it to existing results. The theoretical analysis of convergence and optimality of the proposed algorithm is shown in Section IV. In Section V, we present simulation results to verify the efficacy of our algorithm, and a case study on building energy management. Section VI concludes this paper and provides some directions for future work.

Notation: $\mathbb{R}$ and $\mathbb{N}$ denote the sets of real numbers and natural numbers (excluding zero), respectively. $\mathbb{R}^n$ denotes the n-dimensional Euclidean space. Let $\|\cdot\|_p$ denote the l_p-norm operator. A sequence of elements in $\mathbb{R}^n$ is represented by $(x(k))_{k\in\mathbb{N}}$. For a set $X \subset \mathbb{R}^n$, denote its convex hull by $\mathrm{conv(X)}$. The subdifferential (i.e., the set of all subgradients) of function f at $x \in \mathbb{R}^n$ is denoted by $\partial f(x)$. If f is differentiable, then its gradient is denoted by $\nabla f(x)$. For a matrix A, A_{ij} denotes its (ij)-th element. Denote by $\mathrm{dist}(y, \mathcal{X})$ the Euclidean distance of a vector y from a set $\mathcal{X}$, i.e., $\mathrm{dist}(y, \mathcal{X}) = \inf_{x\in\mathcal{X}}\|y - x\|$.

II. Constrained Optimization

In this section, we formulate a constrained optimization problem and specify some assumptions adopted throughout this paper. We next review heavy-ball methods from a centralized point of view.

A. Problem Set-Up

Consider the following optimization problem of multiple agents $\mathcal{N} \equiv \{1, \cdots, N\}$:

Problem 1:

$$\begin{aligned} \min_{x} \ \ & f(x) = \sum_{i\in\mathcal{N}} f_i(x) \\ \text{subject to } & x \in \cap_{i=1}^{N}\mathcal{X}_i, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ represents the global decision variable and each $\mathcal{X}_i \subset \mathbb{R}^n$ is the local constraint set of agent i, $i \in \mathcal{N}$. Each objective function $f_i : \mathbb{R}^n \to \mathbb{R}$ is convex, and only known to agent $i \in \mathcal{N}$. Note that we do not assume differentiability of f_i which is allowed to be non-smooth. All agents aim to collaboratively minimize the summation of all local objective functions while agreeing on a common value denoted by x^* for the decision vector $x \in \mathbb{R}^n$, where x^* denotes an optimal solution of Problem 1 such that $f(x^*) \le f(x)$ for all $x \in \cap_{i=1}^{N}\mathcal{X}_i$.

We impose the following assumption throughout the paper.

Assumption 2.1: We assume that:

(i) The set $\mathcal{X}_i$ is compact and convex, for all $i \in \mathcal{N}$, and $\cap_{i=1}^{N}\mathcal{X}_i$ has a non-empty interior.

(ii) For all $i \in \mathcal{N}$, the subgradient $g_i \in \partial f_i(x)$ of f_i is bounded on $\mathrm{conv}(\cup_{i=1}^{N}\mathcal{X}_i)$, i.e, there exists $L \in (0, \infty)$ such that

$$\|g_i\|_2 \le L, \quad \forall i \in \mathcal{N}. \tag{2}$$

In Assumption 2.1, item (i) ensures that the optimal solution set of Problem 1 is non-empty. Item (ii) is common in the literature and is satisfied by many functions such as piecewise-linear functions, quadratic functions and logistic-regression functions [16], [17], [20], [28]. In [28], the authors provided a technical condition on the domain of functions f_i, $i \in \mathcal{N}$, whose satisfaction acts as a sufficient condition for item (ii).

B. The Heavy-Ball Method

This subsection reviews the heavy-ball method, on which the proposed algorithm is based. We firstly consider the unconstrained optimization problem $\min_{x\in\mathbb{R}^n} f(x)$ which is μ-strongly convex. The heavy-ball method [32] is an iterative scheme that involves the following update rule

$$x(k+1) = x(k) - \alpha\nabla f(x(k)) + \beta\big(x(k) - x(k-1)\big), \tag{3}$$

where k denotes the iteration index. The last term $\beta\big(x(k) - x(k-1)\big)$ is the momentum term, which is related to the past iterate information with momentum parameter $\beta \in [0, 1)$ and $x(0) = x(1)$. Under a proper choice of step-size α and parameter β, the heavy-ball method could achieve a local accelerated convergence of $\mathcal{O}\big((\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}})^k\big)$ which is faster than the gradient descent method ($\beta = 0$) with convergence rate $\mathcal{O}\big((\frac{L-\mu}{L+\mu})^k\big)$, where L denotes the Lipschitz continuity constant of f. The global linear convergence of this method is developed in [7], [33], [34].

Recently, several papers have studied the heavy-ball method to solve the constrained non-smooth optimization problem $\min_{x\in\mathcal{X}} f(x)$. In [9], this method is naturally encoded by the following update rule

$$x(k+1) = P_{\mathcal{X}}\big[x(k) - \alpha(k)g(k) + \beta(k)\big(x(k) - x(k-1)\big)\big], \tag{4}$$

where $P_{\mathcal{X}}[\cdot]$ represents the projection on the constraint set $\mathcal{X}$, and $g(k)$ is any subgradient of f evaluated at $x(k)$. Note that the momentum parameter $\beta(k) \in [0, 1)$ is iteration-varying. In [9], the authors explore the effects of the heavy-ball momentum on acceleration of convergence. Under certain conditions of $\alpha(k)$ and $\beta(k)$, the iteration above can achieve an optimal convergence rate $\mathcal{O}(\frac{1}{\sqrt{k}})$ for non-smooth problems.

Note that the above heavy-ball methods are centralized. In the next section we will extend them to achieve a novel distributed heavy-ball algorithm to solve Problem 1. Unlike existing distributed heavy-ball methods in [10]–[13], we consider different constraint sets per agent over a time-varying communication network.

III. Distributed Heavy-Ball Algorithm

A. Distributed Computation Framework

In this paper, we aim at solving Problem 1 over a time-varying communication network wherein each agent can only exchange its private information such as the current estimate for the optimal solution $x_i(k), i \in \mathcal{N}$, with its neighbours at a given iteration k. The communication topology of the multi-agent system could be described by an undirected graph denoted by $\mathcal{G}(k) \triangleq \langle\mathcal{N}, \mathcal{E}(k)\rangle$, where $\mathcal{N}$ and $\mathcal{E}(k)$ denote the set of agents (vertex set) and the edge set at iteration k, respectively. Each edge $(i, j) \in \mathcal{E}(k)$ represents the communication between node i and node j. The weight matrix is denoted by $A(k)$ with $A_{ij}(k) > 0$ when $(i, j) \in \mathcal{E}(k)$ and $A_{ij}(k) = 0$

otherwise. Furthermore, we define $\mathcal{G}(\infty) \triangleq \langle \mathcal{N}, \mathcal{E}(\infty) \rangle$ with $(i,j) \in \mathcal{E}(\infty)$ representing that node i could communicate with its neighbour j infinitely often. We have the following assumption on $A(k)$, which has been widely considered in the distributed optimization literature, e.g., [1], [16], [27], [28].

Assumption 3.1: We assume that:
(i) The graph $(\mathcal{N}, \mathcal{E}_\infty)$ is connected. There exists $T \geq 1$ such that agent i receives the information sent by j at least once in every consecutive T iterations.
(ii) There exists $\eta \in (0,1)$ such that $A_{ii}(k) \geq \eta$ for all $k \in \mathbb{N}$, $i,j \in \mathcal{N}$, and if $A_{ij}(k) > 0$ we then have $A_{ij}(k) \geq \eta$.
(iii) Matrix $A(k)$ is doubly stochastic for each k, i.e., $\sum_{i=1}^N A_{ij}(k) = 1$ for all $j \in \mathcal{N}$, and $\sum_{j=1}^N A_{ij}(k) = 1$, for all $i \in \mathcal{N}$.

B. Proposed Algorithm

In this section, we present a distributed heavy-ball method such that each agent could update its own decision vector copy simultaneously. In fact, each agent has a local copy for the global variable and a corresponding subgradient to cooperate with its neighbours at each iteration step, and then all agents eventually agree on a common decision variable. The main steps of the proposed algorithm are summarized in Algorithm 1.

Algorithm 1 Distributed heavy-ball algorithm.

1: **Initialization** $x_i(0) \in \mathbb{R}^n$, $g_i(0) \in \partial f_i(x_i(0))$, $s_i(0) = g_i(0)$, $i \in \mathcal{N}$;
For each $i \in \mathcal{N}$, **repeat until convergence**
2: $g_i(k) \in \partial f_i(x_i(k))$;
3: $s_i(k) = \sum_{j=1}^N A_{ij}(k-1)s_j(k-1) + g_i(k) - g_i(k-1)$;
4: $z_i(k) = \sum_{j=1}^N A_{ij}(k)x_j(k)$;
5: $x_i(k+1) = \mathcal{P}_{\mathcal{X}_i}\big[z_i(k) - \alpha(k)s_i(k) + \beta(x_i(k) - x_i(k-1))\big]$;
6: $k \leftarrow k+1$.
end

We first initialize the decision vector $x_i(0)$, for all $i \in \mathcal{N}$ which can be chosen arbitrarily, and calculate the corresponding subgradient $g_i(0)$ evaluated at $x_i(0)$. Note that $s_i(0) = g_i(0)$ is a necessary condition to show the convergence of Algorithm 1, and has been widely considered in consensus-based algorithms, e.g., [11], [14], [17], [35].

At iteration step k, each agent i calculates its weighted subgradient $s_i(k)$ (Step 3) according to the dynamic average consensus mechanism proposed in [36] that involves the local information and neighbours' gradient information at previous iteration step. This average consensus scheme is used to track the subgradient of the aggregated objective function, i.e., to estimate the global subgradient of Problem 1. Meanwhile, agent i receives information of tentative decision variables from its neighbours and averages them through the weight matrix $A(k)$ such that the auxiliary variable $z_i(k), \forall i \in \mathcal{N}$ could be obtained (Step 4). Finally, motivated by aforementioned heavy-ball methods in (3) and (4), we update the primal variable $x_i(k), \forall i \in \mathcal{N}$ through the gradient projection on local constraint $\mathcal{X}_i$ in Step 5, which includes the momentum term $\beta\big(x_i(k) - x_i(k-1)\big)$ using the past iterate information. In this iteration step, β and $\alpha(k)$ are the fixed momentum parameter and time-varying step-size, respectively. Under certain conditions, Algorithm 1 converges to a global optimum; this is shown in Section IV.

C. Relation to Existing Results

The existing distributed methods to solve the consensus optimization problems mainly involve the update of local estimates for the global variable and a local gradient update. These two points are also the primary distinctions between the existing methods. If we do not consider the momentum term, i.e. set $\beta = 0$, Algorithm 1 can be roughly simplified to several existing methods that employ gradient averaging, such as these in [21], [28], [37].

Specifically, [37] proposed a gradient-tracking based scheme known as DIGing algorithm to solve Problem 1 without constraints. The updates of the local gradient $s_i(k)$ and estimate $x_i(k+1)$ are the same as Step 3 and Step 5 in Algorithm 1 without projection and momentum term, respectively. Furthermore, the authors in [21] extend the results in [37] to constrained cases. A distributed proximal-tracking algorithm is proposed to solve Problem 1 by combining the DIGing algorithm with a proximal-minimization algorithm, which is written compactly as

$$\begin{aligned} x_i(k+1) &= \sum_{j=1}^N A_{ij}x_j(k) - \alpha s_i(k+1), \\ s_i(k+1) &= \sum_{j=1}^N A_{ij}s_j(k) + v_i(x_i(k+1)) - v_i(x_i(k)), \end{aligned}$$

where the update of local gradient s_i is the same as Step 3 in Algorithm 1 and v_i is the subgradient of the summation of local objective function f_i and the corresponding indicator function of the local constraint set. Note that the local estimate $x_i(k+1)$ is obtained based on the subgradient calculated at $x_i(k+1)$ rather than $x_i(k)$. This update makes it possible to adopt a constant step-size when solving a constrained non-smooth problem. In order to make it implementable, several auxiliary variables are introduced in [21] such that each agent can make decisions locally. However, [37] does not consider a time-varying communication network.

Closely related algorithm to Algorithm 1 without the momentum term is the one in [28] which involves subgradient averaging and projection. The main update steps involve

$$x_i(k+1) = \mathcal{P}_{\mathcal{X}_i}\big[z_i(k) - \alpha(k)s_i(k)\big], \tag{6a}$$

$$s_i(k+1) = \sum_{j=1}^N A_{ij}g_j(z_j(k+1)), \tag{6b}$$

where $g_i(\cdot) \in \partial f_i(\cdot)$ is a subgradient of f_i, and $z_i(k+1) = \sum_{j=1}^N A_{ij}x_j(k+1)$. Note that the updates of the weighted subgradient $s_i(k+1)$ are different in Algorithm 1 and in (6). In particular, the dynamic average consensus mechanism in

Algorithm 1 involves the subgradient information at all previous times, while in (6), the subgradient is only related to the information at iteration step k. In addition, $s_i(k+1)$ in (6) is calculated after obtaining the primal decision variables of all its neighbours, while in Algorithm 1, it is calculated using only local information at iteration $k+1$ and pre-known history information, which enables less amount of information to be transfered and fewer communication rounds compared to (6). In the numerical investigations of Section V, we also show that Algorithm 1 exhibits a faster convergence than the iterations above proposed in [28].

IV. Convergence Analysis

This section provides a convergence and optimality analysis for Algorithm 1. We start by summarizing the main results, and then provide some fundamental auxiliary lemmas which are instrumental for the convergence proof. Finally, we provide the proofs of the main results.

A. Statement of Main Results

We impose the following assumption on the step-size $\alpha(k)$.

Assumption 4.1: Suppose that the sequence $(\alpha(k))_{k\in\mathbb{N}}$ adopted in Algorithm 1 satisfies the following properties:
(i) $\alpha(k)$ is non-negative and non-increasing;
(ii) $\sum_{k=1}^{\infty}\alpha(k)=\infty$, and $\sum_{k=1}^{\infty}\alpha(k)^2<\infty$.

Theorem 4.1: Let $(x_i(k))_{k\in\mathbb{N}}, \forall i\in\mathcal{N}$ be the sequences generated by Algorithm 1. Under Assumptions 2.1, 3.1 and 4.1, there exists $\eta_1\in(0,\infty)$ such that for any $\beta\in(0,\frac{1}{18\eta_1+9})$, there exists an optimal solution x^* of Problem 1 such that,

$$\lim_{k\to\infty}\|x_i(k)-x^*\|_2=0,\quad \forall i\in\mathcal{N}. \tag{7}$$

Remark: The value of momentum parameter β has an upper bound less than 1. The above range is just a sufficient condition for convergence. The algorithm may still converge with β outside this range, which is verified in simulations of Section V. The exact value of parameter $\eta_1>0$ is related to the total number of agents, the properties of the underlying optimization problem and the connectivity of the communication network. We provide its value after (21) in Lemma 4.4, once some mathematical preliminaries are provided.

Before stating the convergence rate, we first define an averaged sequence:

$$\widehat{x}_i(k)=\frac{1}{S(k)}\sum_{r=1}^{k}\alpha(r)x_i(r) \tag{8}$$

where $S(k)=\sum_{r=1}^{k}\alpha(r)$, and $(x_i(k))_{k\in\mathbb{N}}, \forall i\in\mathcal{N}$ are the sequences obtained from Algorithm 1 with $\widehat{x}_i(0)=x_i(0)$.

Theorem 4.2: Under Assumptions 2.1 and 3.1, for any $\beta\in(0,\frac{1}{18\eta_1+9})$ and for $\alpha(k)=\frac{\sigma}{\sqrt{k+1}}$ with $\sigma>0$, we have that
(i) The sequence $\left(\|\widehat{x}_i(k)-\widehat{x}_j(k)\|_2\right)_{k\in\mathbb{N}}$ converges to zero at rate $\mathcal{O}(\frac{\ln k}{\sqrt{k}})$ for all $i,j\in\mathcal{N}$;
(ii) The sequence $\left(|\sum_{i=1}^{N}f_i(\widehat{x}_i(k))-f(x^*)|\right)_{k\in\mathbb{N}}$ converges to zero at rate $\mathcal{O}(\frac{\ln k}{\sqrt{k}})$.

Theorem 4.2 establishes a convergence rate in objective value along the weighted running average $\widehat{x}_i(k)$. Such an auxiliary variable computed in a recursive fashion has been widely introduced in the literature, which exhibits superior convergence properties than $x_i(k)$ [24], [28], [38]. Note that the similar convergence rates can also be derived under more general step-size choices, e.g., $\alpha(k)=\frac{1}{k^a}$ with $a\in[0.5,1)$.

B. Mathematical Preliminaries

We define the following variables for each $i\in\mathcal{N}$:

$$v(k)=\frac{1}{N}\sum_{i=1}^{N}x_i(k),\quad \bar{s}(k)=\frac{1}{N}\sum_{i=1}^{N}s_i(k), \tag{9}$$

$$\bar{v}(k)=\frac{\rho}{\epsilon(k)+\rho}v(k)+\frac{\epsilon(k)}{\epsilon(k)+\rho}\bar{x}, \tag{10}$$

$$e_i(k+1)=x_i(k+1)-z_i(k),\quad \forall k\geq 0 \tag{11}$$

where $v(k)$ and $\bar{s}(k)$ represent the average of agents' estimates and the average of their weighted subgradients at time k, respectively. The point $\bar{x}$ is in the interior of the feasible set $\cap_{i=1}^{N}\mathcal{X}_i$ (which is non-empty under Assumption 2.1), with the 2-norm ball of centre $\bar{x}$ and radius $\rho>0$ contained in $\cap_{i=1}^{N}\mathcal{X}_i$, and $\epsilon(k)=\sum_{i=1}^{N}\text{dist}(v(k),\mathcal{X}_i)$. As shown in [28], even though $x_i(k)$ and $v(k)$ do not necessarily belong to $\cap_{i=1}^{N}\mathcal{X}_i$, we always have that $\bar{v}(k)\in\cap_{i=1}^{N}\mathcal{X}_i$. The error $e_i(k+1)$ denotes the difference of the local estimate $x_i(k+1)$ from its weighted value $z_i(k)$ computed at k by agent i.

Lemma 4.1: Consider Algorithm 1 and Assumption 3.1. If $s_i(0)=g_i(0)$, $i\in\mathcal{N}$, then $\sum_{i=1}^{N}s_i(k)=\sum_{i=1}^{N}g_i(k)$, for all $k\in\mathbb{N}$.

Proof: It can be immediately obtained by summing the equality shown in Step 3 of Algorithm 1 for all $i\in\mathcal{N}, k=1,2,\cdots$ and applying the doubly stochastic property of $A(k)$ due to Assumption 3.1 and the fact $s_i(0)=g_i(0)$. ■

Lemma 4.2: Under Assumptions 2.1 and 3.1, the weighted subgradient $s_i(k)$ of function $f_i(x)$ is bounded for all $k\in\mathbb{N}$, such that

$$\|s_i(k)\|_2\leq\widehat{L},\quad \forall i\in\mathcal{N}, \tag{12}$$

where $\widehat{L}\equiv\lambda NL+\frac{2\lambda NL}{1-q}+5L$ with $\lambda=2\left(1+\eta^{-(N-1)T}\right)/\left(1-\eta^{(N-1)T}\right)\in\mathbb{R}_+$ and $q=\left(1-\eta^{(N-1)T}\right)^{\frac{1}{(N-1)T}}\in(0,1)$.

Proof: We first define the subgradient error $E_i(k+1)\triangleq g_i(k+1)-g_i(k)$, and recall the update of s_i in Algorithm 1, for all $i\in\mathcal{N}$. Then,

$$s_i(k+1)=\sum_{j=1}^{N}A_{ij}(k)s_j(k)+E_i(k+1). \tag{13}$$

Consider Assumption 3.1 and $(\bar{s}(k))_{k\in\mathbb{N}}$ defined in (9). By Lemma 2 of [27], we directly obtain the following inequality

$$\begin{aligned}\|s_i(k+1)-\bar{s}(k+1)\|_2\leq\ &\lambda q^k\sum_{j=1}^{N}\|s_j(0)\|_2+\|E_i(k+1)\|_2\\&+\sum_{r=0}^{k-1}\lambda q^{k-r-1}\sum_{j=1}^{N}\|E_j(r+1)\|_2+\frac{1}{N}\sum_{j=1}^{N}\|E_j(k+1)\|_2.\end{aligned} \tag{14}$$

By Assumption 2.1, it can be obtained that $\|s_i(0)\|_2=\|g_i(0)\|_2\leq L$, and $\|E_i(k+1)\|_2\leq 2L$ due to the triangle

inequality. Therefore, based on (14), we have

$$\|s_i(k+1)-\bar{s}(k+1)\|_2 \le \lambda NL + 2\lambda NL\sum_{r=0}^{k-1} q^{k-r-1} + 4L \le \lambda NL + \frac{2\lambda NL}{1-q} + 4L, \quad (15)$$

where the second inequality holds due to the relation $\sum_{r=0}^{k-1} q^{k-r-1} < \sum_{r=0}^{\infty} q^r = \frac{1}{1-q}$.

Furthermore, $\|\bar{s}(k+1)\|_2 = \frac{1}{N}\|\sum_{i=1}^N g_i(k+1)\|_2$ holds by Lemma 4.1. Since we have $\|g_i(k+1)\|_2 \le L$ by Assumption 2.1 and the relation $\|a+b\|_2 \le \|a\|_2 + \|b\|_2$, it can be obtained that $\|\bar{s}(k+1)\|_2 \le L$. Hence we can derive the following inequality by (15):

$$\|s_i(k+1)\|_2 \le \|s_i(k+1)-\bar{s}(k+1)\|_2 + \|\bar{s}(k+1)\|_2 \le \lambda NL + \frac{2\lambda NL}{1-q} + 5L,$$

for all $k \in \mathbb{N}$. Setting $\widehat{L} = \lambda NL + \frac{2\lambda NL}{1-q} + 5L$ leads to (12), thus concluding the proof. ∎

Lemma 4.3: Consider any scalar sequences $(p(r))_{r\in\mathbb{N}}$ and non-negative parameter sequences $(\theta(r))_{r\in\mathbb{R}}$ with $\sum_{r=1}^k \theta(r) \le 1$. For any $k \ge 1$ we have the following relation:

$$\big(\sum_{r=1}^k \theta(r)p(r)\big)^2 \le \sum_{r=1}^k \theta(r)p(r)^2. \quad (16)$$

Proof: When $\sum_{r=1}^k \theta(r) = 1$, the above inequality holds due to the convexity of function $(\cdot)^2$. When $\sum_{r=1}^k \theta(r) < 1$, calculating the derivative of $\big(\sum_{r=1}^k \theta(r)p(r)\big)^2 - \sum_{r=1}^k \theta(r)p(r)^2$ with respect to p, we get that its maximum value is equal to zero and is achieved at $p(r) = 0$, $r = 1,\dots,k$. Hence, $\big(\sum_{r=1}^k \theta(r)p(r)\big)^2 - \sum_{r=1}^k \theta(r)p(r)^2 \le 0$, thus concluding the proof. ∎

The relation established in Lemma 4.3 is crucial for the proof of Lemma 4.4 below.

Lemma 4.4: Consider Assumptions 2.1 and 3.1 and let $(x_i(k))_{k\in\mathbb{N}}, \forall i \in \mathcal{N}$ be the sequences generated by Algorithm 1. We have that

(i)

$$\sum_{i=1}^N \|x_i(k)-\bar{v}(k)\|_2 \le \mu\sum_{i=1}^N \|x_i(k)-v(k)\|_2, \quad (17)$$

$$\sum_{i=1}^N \|x_i(k)-\bar{v}(k)\|_2^2 \le N\mu^2\sum_{i=1}^N \|x_i(k)-v(k)\|_2^2 \quad (18)$$

where $\mu = \frac{2}{\rho}ND+1$, and D is the diameter of the set $\cup_{i=1}^N X_i$.

(ii) for any $\bar{L}, \widetilde{L} > 0$ and any $\xi_1, \hat{\xi}_1 \in (0, \frac{1}{2})$,

$$2\bar{L}\sum_{k=1}^K \alpha(k)\sum_{i=1}^N \|x_i(k+1)-\bar{v}(k+1)\|_2 < \xi_1\sum_{k=1}^K\sum_{i=1}^N \|e_i(k+1)\|_2^2 + \xi_2\sum_{k=1}^K \alpha(k)^2 + \xi_3, \quad (19)$$

$$2\widetilde{L}\sum_{k=1}^K \alpha(k)\sum_{i=1}^N \|x_i(k)-\bar{v}(k)\|_2 < \hat{\xi}_1\sum_{k=1}^K\sum_{i=1}^N \|e_i(k)\|_2^2 + \hat{\xi}_2\sum_{k=1}^K \alpha(k)^2 + \hat{\xi}_3, \quad (20)$$

$$\sum_{k=1}^K\sum_{i=1}^N \|x_i(k)-\bar{v}(k)\|_2^2 < \eta_1\sum_{k=1}^K\sum_{i=1}^N \|e_i(k)\|_2^2 + \eta_2, \quad (21)$$

where $\xi_2, \hat{\xi}_2$ and $\xi_3, \hat{\xi}_3$ are positive constants, and $\eta_1 = \frac{4N^3\mu^2\lambda^2}{(1-q)^2} + 8N\mu^2, \eta_2 = \frac{4N^4D^2\mu^2\lambda^2}{1-q^2}$ with λ and q defined in Lemma 4.2.

Proof: The derivation of (17)-(20) follows directly from Lemmas 2, 3 of [27], and is omitted in the interest of space.

Establishing (21) offers a relationship between $\|x_i(k)-\bar{v}(k)\|_2^2$ and $\|e_i(k)\|_2^2$, which is typically instrumental in convergence analyses for similar algorithms. However, this is challenging in our context. To this end, we provide a novel proof-line to establish (21), which extends the results in [27], [28].

We first consider Assumption 3.1, $(v(k))_{k\in\mathbb{N}}$ defined in (9), and $x_i(k+1) = \sum_{j=1}^N A_{ij}(k)x_j(k) + e_i(k+1)$ in (11). Then by Lemma 2 of [27], we have the following inequality

$$\|x_i(k+1)-v(k+1)\|_2 \le \lambda q^k\sum_{j=1}^N \|x_j(0)\|_2 + \|e_i(k+1)\|_2 + \sum_{r=0}^{k-1}\lambda q^{k-r-1}\sum_{j=1}^N \|e_j(r+1)\|_2 + \frac{1}{N}\sum_{j=1}^N \|e_j(k+1)\|_2, \quad (22)$$

which is similar with (14).

Then the following relation could be obtained by squaring both sides of (22) and applying the triangle inequality:

$$\|x_i(k+1)-v(k+1)\|_2^2 \le 4N\lambda^2\sum_{j=1}^N\Big[\sum_{r=0}^{k-1} q^{k-r-1}\|e_j(r+1)\|_2\Big]^2 + 4\|e_i(k+1)\|_2^2 + 4N\lambda^2q^{2k}\sum_{j=1}^N \|x_j(0)\|_2^2 + \frac{4}{N}\sum_{j=1}^N \|e_j(k+1)\|_2^2. \quad (23)$$

Note that if we adopt the triangle inequality to bound the first term on the right-hand side of (23), giving rise to $\|e_j(r+1)\|_2^2$, then its upper bound will relate to iteration step k. As the algorithm proceeds, i.e., $k \to \infty$, the calculated upper bound of $\|x_i(k+1)-v(k+1)\|_2^2$ will also tend to infinity, thus not being useful to establish our claim. Therefore, we take advantage of the fact that $q \in (0,1)$ to handle this square term. It is known that $(1-q)\sum_{r=0}^{k-1} q^{k-r-1} = 1-q^k \le 1$.

Then by applying Lemma 4.3, we can obtain that

$$\Big(\sum_{r=0}^{k-1} q^{k-r-1}\left\|e_j(r+1)\right\|_2\Big)^2 \leq \frac{1}{1-q}\sum_{r=0}^{k-1} q^{k-r-1}\left\|e_j(r+1)\right\|_2^2.$$

Combining it with (23) and (18) leads to the following inequality:

$$\begin{aligned}\sum_{k=1}^{K}\sum_{i=1}^{N}\|x_i(k)-\bar{v}(k)\|_2^2 &\leq 4N^3\mu^2\lambda^2\sum_{k=1}^{K}q^{2(k-1)}\sum_{i=1}^{N}\|x_i(0)\|_2^2\\ &+8N\mu^2\sum_{k=1}^{K}\sum_{i=1}^{N}\|e_i(k)\|_2^2\\ &+\frac{4N^3\mu^2\lambda^2}{1-q}\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{r=0}^{k-2}q^{k-r-2}\|e_i(r+1)\|_2^2. \end{aligned}\tag{24}$$

Next we will analyze the first term and last term on the right-hand side of (24) separately. Under Assumption 2.1, we know that $\|x_i(0)\|_2^2 \leq D^2, \forall i \in \mathcal{N}$. Hence, $\sum_{i=1}^{N}\|x_i(0)\|_2^2 \leq ND^2$ which gives

$$\begin{aligned}&4N^3\mu^2\lambda^2\sum_{k=1}^{K}q^{2(k-1)}\sum_{i=1}^{N}\|x_i(0)\|_2^2\\ &\leq 4N^4D^2\mu^2\lambda^2\sum_{k=1}^{K}q^{2(k-1)}\\ &<\frac{4N^4D^2\mu^2\lambda^2}{1-q^2}\end{aligned}\tag{25}$$

where the last inequality holds due to $\sum_{k=1}^{\infty}q^{2(k-1)}=\frac{1}{1-q^2}$.

In terms of the last term of (24), we have

$$\begin{aligned}&\frac{4N^3\mu^2\lambda^2}{1-q}\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{r=0}^{k-2}q^{k-r-2}\|e_i(r+1)\|_2^2\\ &=\frac{4N^3\mu^2\lambda^2}{1-q}\sum_{i=1}^{N}\sum_{r=0}^{K-2}\|e_i(r+1)\|_2^2\sum_{t=0}^{K-r-2}q^t\\ &<\frac{4N^3\mu^2\lambda^2}{1-q}\sum_{i=1}^{N}\sum_{r=0}^{K-2}\|e_i(r+1)\|_2^2\sum_{t=0}^{\infty}q^t\\ &<\frac{4N^3\mu^2\lambda^2}{(1-q)^2}\sum_{i=1}^{N}\sum_{k=1}^{K}\|e_i(k)\|_2^2\end{aligned}\tag{26}$$

where the first equality follows from the series convolution, and the last inequality holds due to $\sum_{t=0}^{\infty}q^t=\frac{1}{1-q}$ and a summation index change from r to k.

Substituting (25) and (26) into (24) leads to (21), thus concluding the proof. ∎

Lemma 4.5: Consider Assumptions 2.1 and 3.1 and let $(x_i(k))_{k\in\mathbb{N}}, \forall i \in \mathcal{N}$ be the sequences generated by Algorithm 1, and x^* be an optimal solution of Problem 1. We have that for any parameter $\gamma \in (0,1)$,

$$\begin{aligned}&\sum_{i=1}^{N}\big[\,\|x_i(k+1)-x^*\|_2^2-\beta\|x_i(k)-x^*\|_2^2\big]\\ &\leq\sum_{i=1}^{N}\big[\,\|x_i(k)-x^*\|_2^2-\beta\|x_i(k-1)-x^*\|_2^2\big]\\ &+2(L+\widehat{L})\alpha(k)\sum_{i=1}^{N}\|x_i(k+1)-\bar{v}(k+1)\|_2\\ &+6L\alpha(k)\sum_{i=1}^{N}\|x_i(k)-\bar{v}(k)\|_2+6\beta\sum_{i=1}^{N}\|x_i(k)-\bar{v}(k)\|_2^2\\ &+12\beta\sum_{i=1}^{N}\|x_i(k-1)-\bar{v}(k-1)\|_2^2+6\beta\sum_{i=1}^{N}\|e_i(k)\|_2^2\\ &+\frac{NL^2}{\gamma}\alpha(k)^2-2\alpha(k)\sum_{i=1}^{N}\big(f_i(\bar{v}(k))-f_i(x^*)\big)\\ &-(1-\gamma-3\beta)\sum_{i=1}^{N}\|e_i(k+1)\|_2^2.\end{aligned}\tag{27}$$

Proof: The proof is provided in the Appendix. ∎

The relation established in Lemma 4.5 can be used to prove the following lemma which reveals the link between $x_i(k)$, its weighted value $z_i(k-1)$ and the averaged value $v(k)$.

Lemma 4.6: Consider Assumptions 2.1, 3.1 and 4.1. For any $\beta \in (0, \frac{1}{18\eta_1+9})$, we have the following statements:

(i) $\sum_{k=1}^{K}\sum_{i=1}^{N}\|e_i(k)\|_2^2<\infty$;

(ii) $\lim_{k\to\infty}\|e_i(k)\|_2=0, \quad \forall i\in\mathcal{N}$;

(iii) $\lim_{k\to\infty}\|x_i(k)-v(k)\|_2=0, \quad \forall i\in\mathcal{N}$.

Proof: Summing (27) in Lemma 4.5 from $k=1$ to $k=K$, and applying item (ii) of Lemma 4.4 with $\bar{L}=L+\widehat{L}$ in (19), and $\widetilde{L}=3L$ in (20), we obtain

$$\begin{aligned}&\big(1-\gamma-\xi_1-3\beta\big)\sum_{k=1}^{K}\sum_{i=1}^{N}\|e_i(k+1)\|_2^2\\ &-(\hat{\xi}_1+6\beta\eta_1+6\beta)\sum_{k=1}^{K}\sum_{i=1}^{N}\|e_i(k)\|_2^2\\ &-12\beta\eta_1\sum_{k=1}^{K}\sum_{i=1}^{N}\|e_i(k-1)\|_2^2-\sum_{k=1}^{K}\big(\frac{NL^2}{\gamma}+\xi_2+\hat{\xi}_2\big)\alpha(k)^2\\ &-(\xi_3+\hat{\xi}_3+18\beta\eta_2)+2\sum_{k=1}^{K}\alpha(k)\sum_{i=1}^{N}\big(f_i(\bar{v}(k))-f_i(x^*)\big)\\ &+\sum_{k=1}^{K}\sum_{i=1}^{N}\big[\,\|x_i(k+1)-x^*\|_2^2-\beta\|x_i(k)-x^*\|_2^2\big]\\ &\leq\sum_{k=1}^{K}\sum_{i=1}^{N}\big[\,\|x_i(k)-x^*\|_2^2-\beta\|x_i(k-1)-x^*\|_2^2\big].\end{aligned}\tag{28}$$

Since the two terms $\sum_{k=1}^{K}\sum_{i=1}^{N}\|x_i(k+1)-x^*\|_2^2$ and $\sum_{k=1}^{K}\sum_{i=1}^{N}\|x_i(k)-x^*\|_2^2$ in (28) form telescopic series, they could be replaced by $\sum_{i=1}^{N}\|x_i(K+1)-x^*\|_2^2$ and $\sum_{i=1}^{N}\|x_i(1)-x^*\|_2^2$, respectively. We drop all the non-negative square terms on the left-hand side of (28), which

gives

$$\begin{aligned}
&\big(1-\gamma-\xi_1-\hat{\xi}_1-18\beta\eta_1-9\beta\big)\sum_{k=1}^{K}\sum_{i=1}^{N}\|e_i(k)\|_2^2\\
&\quad+2\sum_{k=1}^{K}\alpha(k)\sum_{i=1}^{N}\big(f_i(\bar{v}(k))-f_i(x^*)\big)\\
&\le\big(1-\gamma-\xi_1-3\beta\big)\sum_{i=1}^{N}\|e_i(1)\|_2^2+12\beta\eta_1\sum_{i=1}^{N}\|e_i(0)\|_2^2\\
&\quad+\sum_{k=1}^{K}\big(\frac{NL^2}{\gamma}+\xi_2+\hat{\xi}_2\big)\alpha(k)^2+(\xi_3+\hat{\xi}_3+18\beta\eta_2)\\
&\quad+\sum_{i=1}^{N}\big[\,\|x_i(1)-x^*\|_2^2+\beta\|x_i(K)-x^*\|_2^2\big]. \qquad (29)
\end{aligned}$$

Note that $\sum_{i=1}^{N}\big(f_i(\bar{v}(k))-f_i(x^*)\big)\ge 0$ due to the optimality of x^*. Therefore, the second term in (29) can also be dropped. If we choose $\beta\in(0,\frac{1}{18\eta_1+9})$ such that $1-\gamma-\xi_1-\hat{\xi}_1-18\beta\eta_1-9\beta>0$, then $\sum_{k=1}^{\infty}\sum_{i=1}^{N}\|e_i(k)\|_2^2$ with $K\to\infty$ is finite due to $(\alpha(k))_{k\in\mathbb{N}}$ being square-summable by Assumption 4.1 and due to the compactness of the feasible set. This yields item (i). The proof of items (ii) and (iii) follows directly from item (i), and is omitted for brevity (see Proposition 3 in [27] for similar developments). ■

Note that for any $\beta\in(0,\frac{1}{18\eta_1+9})$, the parameters $\gamma\in(0,1)$, $\xi_1,\hat{\xi}_1\in(0,\frac{1}{2})$ can be chosen to satisfy the relation $1-\gamma-\xi_1-\hat{\xi}_1-18\beta\eta_1-9\beta>0$ where constant $\eta_1=\frac{4N^3\mu^2\lambda^2}{(1-q)^2}+8N\mu^2$ introduced in (21). For example, one particular choice is that $\gamma=\xi_1=\hat{\xi}_1$ with γ satisfying $1-3\gamma-\beta(18\eta_1+9)>0$.

C. Proof of Main Results

The proofs of Theorem 4.1 and Theorem 4.2 are based on the auxiliary results presented in above Section IV-B. Theorem 4.1 extends the results in [28] by allowing less information exchange and fewer communication rounds, and introducing the momentum term to accelerate the convergence of iterations, which brings challenges to the convergence proof of the algorithm. The proof of Theorem 4.2 is inspired by [28] which also involves the convergence of a running average sequence.

1) Proof of Theorem 4.1: From the proof of Lemma 4.6, we know that $1-\gamma-3\beta\in(0,1)$ for any $\beta\in(0,\frac{1}{18\eta_1+9})$. Therefore we could droop the last term $(1-\gamma-3\beta)\sum_{i=1}^{N}\|e_i(k+1)\|_2^2$ in (27), which leads to the following inequality

$$\varpi(k+1)\le\varpi(k)-\varrho(k)+\varphi(k),$$

where

$$\begin{aligned}
\varpi(k)&=\sum_{i=1}^{N}\big[\,\|x_i(k)-x^*\|_2^2-\beta\|x_i(k-1)-x^*\|_2^2\big];\\
\varrho(k)&=2\alpha(k)\sum_{i=1}^{N}\big(f_i(\bar{v}(k))-f_i(x^*)\big);
\end{aligned}$$

$$\begin{aligned}
\varphi(k)=\;&8L\alpha(k)\sum_{i=1}^{N}\|x_i(k+1)-\bar{v}(k+1)\|_2+\frac{NL^2}{\gamma}\alpha(k)^2\\
&+6L\alpha(k)\sum_{i=1}^{N}\|x_i(k)-\bar{v}(k)\|_2+6\beta\sum_{i=1}^{N}\|x_i(k)-\bar{v}(k)\|_2^2\\
&+6\beta\sum_{i=1}^{N}\|e_i(k)\|_2^2+12\beta\sum_{i=1}^{N}\|x_i(k-1)-\bar{v}(k-1)\|_2^2.
\end{aligned}$$

The sequences $(\varrho(k))_{k\in\mathbb{N}}$ and $(\varphi(k))_{k\in\mathbb{N}}$ are nonnegative, and $\varpi(k)$ is bounded for all k due to Assumption 2.1. By applying item (ii) in Lemma 4.4 and item (i) in Lemma 4.6, under Assumption 4.1 it follows that $\sum_{k=1}^{\infty}\varphi(k)<\infty$. Therefore, by Lemma 3.4 in [39], we could obtain that the sequence $(\varpi(k))_{k\in\mathbb{N}}$ converges to a finite value and $\sum_{k=1}^{\infty}\varrho(k)<\infty$.

The fact that $\sum_{k=1}^{\infty}\varrho(k)<\infty$ implies that there exists a subsequence of $\big(f(\bar{v}(k))-f(x^*)\big)_{k\in\mathbb{N}}$ that converges to zero. Since $f(x)$ is continuous, there also exists a subsequence of $\big(\|\bar{v}(k)-x^*\|_2\big)_{k\in\mathbb{N}}$ converges to zero. Furthermore, we have $\sum_{i=1}^{N}\|x_i(k)-x^*\|_2\le\sum_{i=1}^{N}\|\bar{v}(k)-x^*\|_2+\sum_{i=1}^{N}\|x_i(k)-\bar{v}(k)\|_2$ which together with item (i) of Lemma 4.4 and item (iii) of Lemma 4.6, imply that there exists a subsequence of $\big(\sum_{i=1}^{N}\|x_i(k)-x^*\|_2\big)_{k\in\mathbb{N}}$ that converges to zero.

Hence, we can find a subsequence of $\big(\sum_{i=1}^{N}\big[\,\|x_i(k)-x^*\|_2^2-\beta\|x_i(k-1)-x^*\|_2^2\big]\big)_{k\in\mathbb{N}}$ that converges to zero. Since the sequence $\big(\sum_{i=1}^{N}\big[\,\|x_i(k)-x^*\|_2^2-\beta\|x_i(k-1)-x^*\|_2^2\big]\big)_{k\in\mathbb{N}}$ is convergent, it has a unique limit point, which must thus be zero, i.e. $\lim_{k\to\infty}\sum_{i=1}^{N}\big[\,\|x_i(k)-x^*\|_2^2-\beta\|x_i(k-1)-x^*\|_2^2\big]=0$. Since $\lim_{k\to\infty}\sum_{i=1}^{N}\|x_i(k)-x^*\|_2^2=\lim_{k\to\infty}\sum_{i=1}^{N}\|x_i(k-1)-x^*\|_2^2$, we have $\lim_{k\to\infty}(1-\beta)\sum_{i=1}^{N}\|x_i(k)-x^*\|_2^2=0$, which gives $\lim_{k\to\infty}\sum_{i=1}^{N}\|x_i(k)-x^*\|_2^2=0$ where $\beta\in(0,1)$. Finally, we conclude that the sequences $\big(\|x_j(k)-x^*\|_2\big)_{k\in\mathbb{N}}$ for $j=1,\dots,N$, converge to zero since $\|x_j(k)-x^*\|_2\le\sum_{i=1}^{N}\|x_i(k)-x^*\|_2$. This concludes the proof.

2) Proof of Theorem 4.2: Let $\widehat{v}(k)=\frac{1}{S(k)}\sum_{r=1}^{k}\alpha(r)\bar{v}(r)$ which is similar to the definition of $\widehat{x}_i(k)$ in (8), where $\bar{v}(r)$ is introduced in (10). Since $\bar{v}(k)\in\cap_{i=1}^{N}\mathcal{X}_i$ (see discussion below (10)), we have that $\widehat{v}(k)$ for all $k\in\mathbb{N}$ is feasible.

Under Assumption 2.1, it can be obtained that the function f_i is Lipschitz continuous over $\mathcal{X}_i$, i.e., $|f_i(x)-f_i(y)|\le L\|x-y\|_2$ for all $x,y\in\mathcal{X}_i,\forall i\in\mathcal{N}$ [40]. Therefore, by using the triangle inequality we have that

$$\begin{aligned}
&\Big|\sum_{i=1}^{N}f_i(\widehat{x}_i(k+1))-f(x^*)\Big| \qquad (30)\\
&\le\; f(\widehat{v}(k+1))-f(x^*)+L\sum_{i=1}^{N}\|\widehat{x}_i(k+1)-\widehat{v}(k+1)\|_2,
\end{aligned}$$

where $f(\widehat{v}(k+1))\ge f(x^*)$ since x^* is an optimal solution.

Next, we analyze the two terms on the right-hand side

of (30). As for the first term, we have

$$f(\widehat{v}(k+1)) - f(x^*) \leq \sum_{r=1}^{k+1} \frac{\alpha(r)}{S(k+1)} f(\bar{v}(r)) - f(x^*), \tag{31}$$

where the inequality follows by the definition of $\widehat{v}(k+1)$ and due to convexity of f.

For any $\beta \in (0, \frac{1}{18\eta_1+9})$, we can always find proper parameters $\gamma \in (0,1)$, $\xi_1, \hat{\xi}_1 \in (0, \frac{1}{2})$ satisfying $1-\gamma-\xi_1-\hat{\xi}_1-18\beta\eta_1-9\beta > 0$. Therefore, the first term on the left-hand side of (29) can be dropped, which leads to the following inequality by replacing k by r, and K by k:

$$\begin{aligned}
&2\sum_{r=1}^{k}\alpha(r)\sum_{i=1}^{N}\big(f_i(\bar{v}(r)) - f_i(x^*)\big) \\
&\leq \big(1-\gamma-\xi_1-3\beta\big)\sum_{i=1}^{N}\|e_i(1)\|_2^2 + 12\beta\eta_1\sum_{i=1}^{N}\|e_i(0)\|_2^2 \\
&+ \sum_{r=1}^{k}\big(\frac{NL^2}{\gamma} + \xi_2 + \hat{\xi}_2\big)\alpha(r)^2 + (\xi_3 + \hat{\xi}_3 + 18\beta\eta_2) \\
&+ \sum_{i=1}^{N}\big[\,\|x_i(1)-x^*\|_2^2 + \beta\|x_i(k)-x^*\|_2^2\big] \\
&\leq\ d_1 + d_2\sum_{r=1}^{k}\alpha(r)^2,
\end{aligned} \tag{32}$$

where $d_2 = \frac{NL^2}{\gamma} + \xi_2 + \hat{\xi}_2$ and $d_1 = 4ND^2(2-\gamma-\xi_1-2\beta+12\beta\eta_1) + \xi_3 + \hat{\xi}_3 + 18\beta\eta_2$ due to $\|x_i(k)-x^*\|_2^2 \leq 4D^2$ and $\|e_i(k)\|_2^2 \leq 4D^2$ for all $k \in \mathbb{N}, i \in \mathcal{N}$, with D defined in Lemma 4.4.

Substituting (32) into (31) gives

$$f(\widehat{v}(k+1)) - f(x^*) \leq \frac{d_1}{2S(k+1)} + d_2\frac{\sum_{r=1}^{k+1}\alpha(r)^2}{2S(k+1)}. \tag{33}$$

Considering now the second term on the right-hand side of (30), we have

$$\begin{aligned}
&L\sum_{i=1}^{N}\|\widehat{x}_i(k+1) - \widehat{v}(k+1)\|_2 \\
&\leq \frac{L}{S(k+1)}\sum_{r=1}^{k+1}\alpha(r)\sum_{i=1}^{N}\|x_i(r) - \bar{v}(r)\|_2 \\
&\leq \frac{1}{S(k+1)}\big[\hat{\xi}_1\sum_{r=1}^{k+1}\sum_{i=1}^{N}\|e_i(r)\|_2^2 + \hat{\xi}_2\sum_{r=1}^{k+1}\alpha(r)^2 + \hat{\xi}_3\big]
\end{aligned} \tag{34}$$

where the first inequality holds due to convexity of $\|\cdot\|$, and the second inequality follows by (20) in Lemma 4.4 with $\widetilde{L} = L/2$.

Similarly to the derivation of (32), we drop the second term (non-negative) on the left-hand side of (29), and obtain

$$\overline{d}\sum_{r=1}^{k}\sum_{i=1}^{N}\|e_i(r)\|_2^2 \leq d_1 + d_2\sum_{r=1}^{k}\alpha(r)^2, \tag{35}$$

where $\overline{d} = 1-\gamma-\xi_1-\hat{\xi}_1-18\beta\eta_1-9\beta$. Substituting (35) into (34) leads to

$$\begin{aligned}
&L\sum_{i=1}^{N}\|\widehat{x}_i(k+1) - \widehat{v}(k+1)\|_2 \\
&\qquad \leq \frac{d_3}{S(k+1)} + d_4\frac{\sum_{r=1}^{k+1}\alpha(r)^2}{S(k+1)}
\end{aligned} \tag{36}$$

where $d_3 = \hat{\xi}_3 + \hat{\xi}_1 d_1/\overline{d}$ and $d_4 = \hat{\xi}_2 + \hat{\xi}_1 d_2/\overline{d}$.

Note that $S(k+1)$ can be lower-bounded as

$$\begin{aligned}
S(k+1) &= \sum_{r=1}^{k+1}\frac{\sigma}{\sqrt{r+1}} \geq \int_2^{k+3}\frac{\sigma}{\sqrt{x}}dx \\
&=\ 2\sigma(\sqrt{k+3} - \sqrt{2}) \geq d_5\sqrt{k+1},
\end{aligned} \tag{37}$$

where $d_5 = \sigma(2-\sqrt{2})$. We also have that

$$\begin{aligned}
\sum_{r=1}^{k+1}\alpha(r)^2 &= \sigma^2\sum_{r=1}^{k+1}\frac{1}{r+1} \leq \sigma^2\sum_{r=1}^{k+1}\frac{1}{r} \\
&\leq \sigma^2\Big(\int_1^{k+1}\frac{1}{x}dx + 1\Big) = \sigma^2\ln(k+1) + \sigma^2.
\end{aligned} \tag{38}$$

Item (ii) in Theorem 4.2 follows then by substituting (33), (36)-(38) into (30). Since $\|\widehat{x}_i(k) - \widehat{x}_j(k)\|_2 \leq \sum_{i=1}^{N}\|\widehat{x}_i(k) - \widehat{v}(k)\|_2 + \sum_{i=1}^{N}\|\widehat{x}_j(k) - \widehat{v}(k)\|_2$ by using the triangle inequality, we can obtain that the sequence $\big(\|\widehat{x}_i(k) - \widehat{x}_j(k)\|_2\big)_{k\in\mathbb{N}}$ converges to zero at a rate $\mathcal{O}(\frac{\ln k}{\sqrt{k}})$ by substituting (37) and (38) into (36). This concludes the proof of item (i) in Theorem 4.2.

V. Case Studies

A. Numerical Example and Comparative Study

In this section, we consider a binary-classification logistic-regression problem with ℓ_1-regularization to demonstrate the performance of our proposed distributed heavy-ball algorithm, i.e.,

$$\min_{x\in\mathcal{X}}\sum_{i=1}^{N}\sum_{j=1}^{M_i}\ln\big[1+\exp(-b_{ij}(a_{ij}^\top w + v))\big] + \lambda\|w\|_1, \tag{39}$$

where the optimization vector is defined as $x = [w^\top, v]^\top$ with $w \in \mathbb{R}^p$ and $v \in \mathbb{R}$. Here, $\lambda > 0$ is a regularization parameter to avoid over-fitting, $a_{ij} \in \mathbb{R}^p$ is a feature vector and $b_{ij} \in \{-1, 1\}$ is the corresponding binary label. Suppose the variable satisfies the constraint $\mathcal{X}_i = \mathcal{X} = \{x \in \mathbb{R}^{p+1} : \|x\|_2 \leq c\}$ with parameter $c > 0$.

In our setting, we consider $N = 30$ agents and each agent has $M_i = 20$ training examples with $p = 20$ features. Each feature vector a_{ij} is sampled independently from a standard normal distribution, and the values of a_{ij}, b_{ij} and λ are given based on the criteria in [41]. The agents' individual functions can be represented as

$$f_i(x) = \sum_{j=1}^{M_i}\ln\big(1+e^{-b_{ij}(a_{ij}^\top w + v)}\big) + \frac{\lambda}{N}\|w\|_1 \tag{40}$$

with local constraint sets $\mathcal{X}_i = \{x \in \mathbb{R}^{p+1} : \|x\|_2 \leq 6\}$ for $i = 1, \cdots, N$. It can be observed that the constraint sets $\mathcal{X}_i$ and local functions f_i, $i = 1, \cdots, N$ satisfy Assumption 2.1. Therefore, we apply Algorithm 1 to obtain an optimal solution

of problem (39) in a fully-distributed manner under the step-size choice $\alpha(k) = \frac{1}{k+1}$.

Consider first a time-varying communication network with independent random sparsity degree $d \in (0, 1)$ at each iteration, in which the number of connections among all agents (nodes) is given by dN^2. For a complete network graph, the number of connections is N^2. Fig. 1 shows the evolution of $\frac{|\sum_{i=1}^{30} f_i(x_i(k)) - f^\star|}{f^\star}$ for Algorithm 1, where $f^\star$ denotes the optimal value computed for the sake of our numerical analysis by means of solving (39) in a centralized manner. As we can see, the proposed distributed algorithm can converge to the optimal solution under proper momentum parameters β, and it converges faster as β increases. Note that the algorithm can still converge even if β exceeds the limit value proposed in Theorem 4.1, which turns out here to be $\beta = 0.3$, since this is only a sufficient condition for convergence.

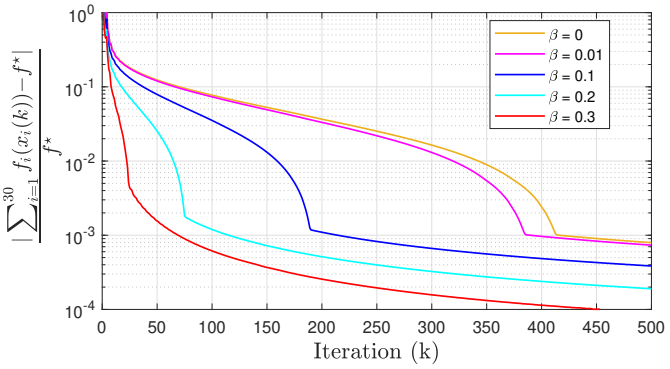
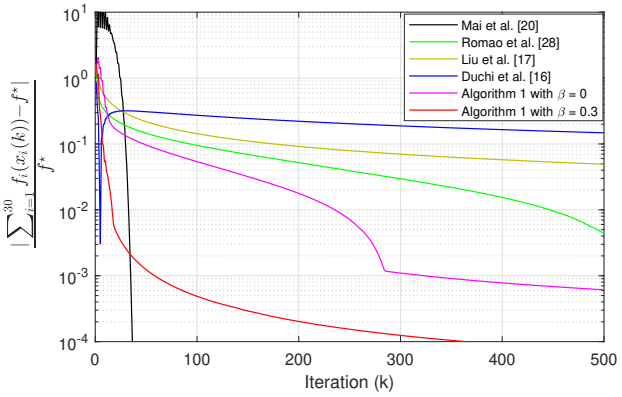
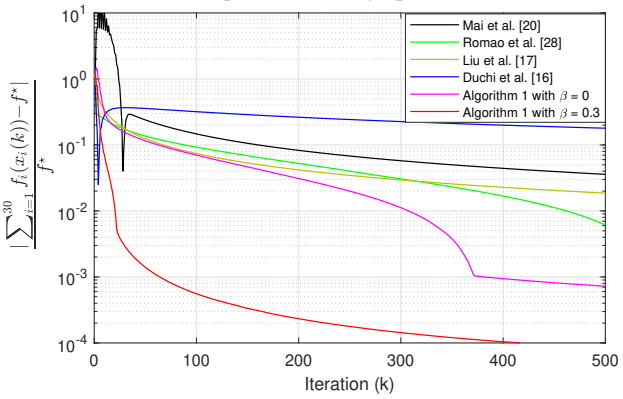


Fig. 1: Evolution of $\frac{|\sum_{i=1}^{30} f_i(x_i(k)) - f^\star|}{f^\star}$ for Algorithm 1 under different momentum parameter values β.

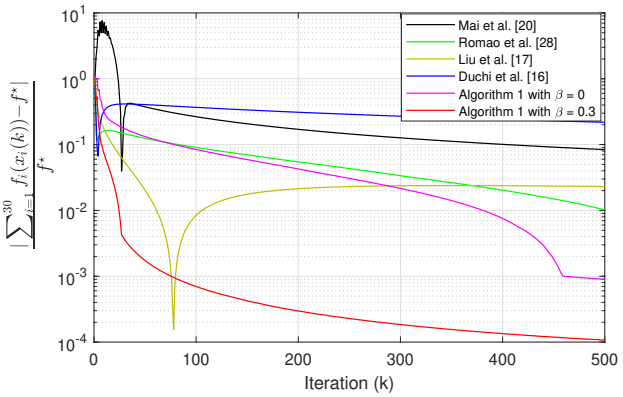
We also compare our algorithm with the following distributed methods in Fig. 2 over a time-invariant communication network: (i) Algorithm 1 in [20] which is a modified version of the projected subgradient method proposed in [25]; (ii) subgradient averaging algorithm in [28]; (iii) subgradient algorithm with double averaging in [17]; (iv) dual averaging algorithm in [16]. They are implemented under four different network connectivity structures: (a) complete network graph; (b) sparse network graph with sparsity degree $d = 0.6$; (c) sparse network graph with $d = 0.3$; (d) line network graph. All settings in these algorithms, such as the step-size, are kept the same as outlined above. It can be observed from Fig. 2 that our algorithm outperforms other ones, especially under a sparse network graph even when the momentum parameter is $\beta = 0$. Only in the case of a complete network graph, our algorithm may not be the fastest; in this specific case (that in practice may not occur due to communication failures), the algorithm in [20] (black line) leads to a faster behaviour. However, it should be mentioned that the algorithm of [20] is supported theoretically only for the case where the communication network is time-invariant, and could not be directly applied to handle the time-varying network that requires different analysis arguments as the ones presented in this work.



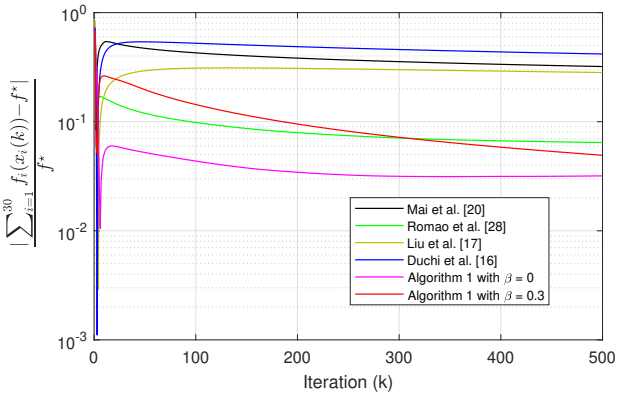
(a) Complete network graph.



(b) Sparse network with $d = 0.6$.



(c) Sparse network with $d = 0.3$.



(d) Line network graph.

Fig. 2: Evolution of $\frac{|\sum_{i=1}^{30} f_i(x_i(k)) - f^*|}{f^*}$ for Algorithm 1 and other related algorithms under different network graphs.

B. Energy Management for a Building District Cooling System

1) Simulation set-up: In this section, we demonstrate the efficacy of the proposed algorithm on a building district cooling system which is composed of multiple buildings and a cooling storage network. Each building is equipped with a chiller plant that can convert electricity into cooling energy. The indoor temperatures of each building could be set within an appropriate range by operating its own chiller. Each building can exchange energy with the cooling storage network which is shared among all buildings in the district, such that the energy utilization efficiency could be improved. In Fig. 3, we show a simple cooling system wherein each building can only exchange information with its neighbours. We aim at coordinating the district cooling energy with individual cooling loads satisfied by minimizing the system total cost over a finite time horizon $\mathcal{T} = \{1, \dots, \widehat{T}\}$.

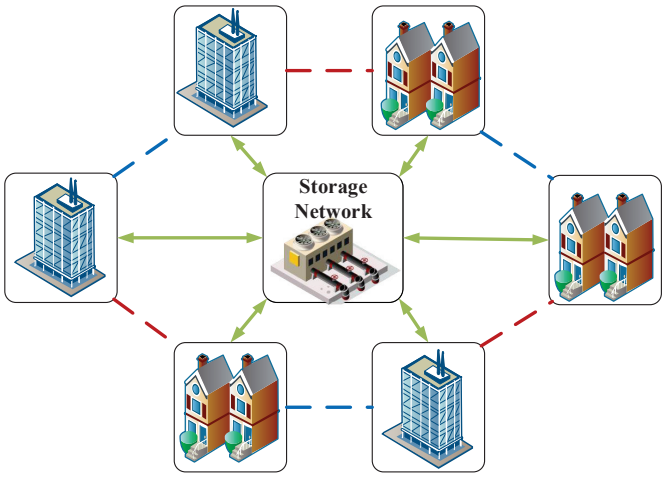


Fig. 3: The structure of a building district cooling system with 6 buildings (chillers), in which dashed lines represent information flows and solid lines represent energy flows.

Consider a district cooling system involving N buildings. Denote by $E_{it}^{\mathrm{B}} \in \mathbb{R}$ the cooling energy request of building $i = 1, \cdots, N$ at $t \in \mathcal{T}$ for temperature regulation. Let $\theta_{it}^{\mathrm{in}}$ and $\theta_{it}^{\mathrm{out}}$ denote the indoor temperature and the outdoor temperature at t, respectively. We adopt the following simplified model to describe the temperature dynamics, which has been widely used in the literature [42], [43]

$$-E_{it}^{\mathrm{B}} = c_i^{\mathrm{air}}(\theta_{it}^{\mathrm{in}} - \theta_{i,t-1}^{\mathrm{in}})/\hat{\tau} - (\theta_{it}^{\mathrm{out}} - \theta_{it}^{\mathrm{in}})/R_i, \tag{41}$$

where parameters c_i^{air}, R_i and $\hat{\tau}$ represent the air specific heat capacity (kWh/°C), the thermal resistance of building i (°C/kW) and the duration of each period (h), respectively. From (41), we can obtain the indoor temperature at t as

$$\theta_{it}^{\mathrm{in}} = (1-a_i)^t \theta_{i,0}^{\mathrm{in}} + \sum_{\tau=1}^{t} (1-a_i)^{t-\tau} \left(a_i \theta_{i\tau}^{\mathrm{out}} - b_i E_{i\tau}^{\mathrm{B}}\right) \tag{42}$$

where $a_i = \frac{1}{c_i^{\mathrm{air}} R_i/\hat{\tau}+1}$, $b_i = \frac{R_i}{c_i^{\mathrm{air}} R_i/\hat{\tau}+1}$, and $\theta_{i,0}^{\mathrm{in}}$ is the initial indoor temperature. We could observe from above models that the current temperature is related to the history temperatures, the current cooling draw, and the outdoor temperature.

Denote by $E_{it}^{\mathrm{ch}} \in \mathbb{R}$ the cooling energy exchange between building i and the energy storage network at t, and $E_{it}^{\mathrm{ch}} > 0$ if building i draws energy from the storage network and $E_{it}^{\mathrm{ch}} < 0$ if i inputs energy to it. The amount of cooling energy stored can be described by a first-order dynamical model [44],

$$E_{t+1}^{\mathrm{stored}} = a E_t^{\mathrm{stored}} - \sum_{i=1}^{N} E_{it}^{\mathrm{ch}}, \tag{43}$$

where coefficient $a \in (0, 1)$ is used to describe energy losses.

Based on the energy balance, the total amount of cooling generation of the chiller in building i at t can be given as:

$$E_{it,\mathrm{c}}^{\mathrm{chiller}} = E_{it}^{\mathrm{B}} - E_{it}^{\mathrm{ch}}. \tag{44}$$

Let $E_{it,\mathrm{e}}^{\mathrm{chiller}}$ represent the electricity needed to produce a certain amount of cooling energy $E_{it,\mathrm{c}}^{\mathrm{chiller}}$. Following [44], we model $E_{it,\mathrm{e}}^{\mathrm{chiller}}$ as a biquadratic convex approximation:

$$E_{it,\mathrm{e}}^{\mathrm{chiller}} = c_{2,i} {E_{it,\mathrm{c}}^{\mathrm{chiller}}}^4 + c_{1,i} {E_{it,\mathrm{c}}^{\mathrm{chiller}}}^2 + c_{0,i}, \tag{45}$$

where parameters $c_{0,i}, c_{1,i}, c_{2,i}$ are related to individual conditions.

The resulting optimization problem is formulated as follows:

$$\min_{\{\{\theta_{it}^{\mathrm{in}}, E_{it}^{\mathrm{ch}} \in \mathbb{R}\}_{i=1}^{N}\}_{t=1}^{\widehat{T}}} \sum_{t=1}^{\widehat{T}} \sum_{i=1}^{N} p_{it} E_{it,\mathrm{e}}^{\mathrm{chiller}} \quad \text{subject to (42)} - \text{(45)}, \ \text{(47)} - \text{(51)} \tag{46}$$

where $p_{it} \in \mathbb{R}$ is the electricity price for building i at t. For each building $i = 1, \dots, N$ at $t \in \mathcal{T}$, the constraints in (47)-(51) are detailed below.

(1) Electricity limits: Due to the chiller size and maximum capability, the electricity drawn from the distribution network is limited which satisfies the following constraint:

$$E_{it,\mathrm{e}}^{\mathrm{chiller}} \le E_{i,\max}, \tag{47}$$

where $E_{i,\max}$ denotes its upper bound.

(2) Cooling energy limits: The cooling energy request E_{it}^{B} is non-negative, i.e.,

$$E_{it}^{\mathrm{B}} \ge 0. \tag{48}$$

(3) Comfort constraints: The individual indoor temperature is within a certain comfort range, i.e.,

$$\theta_{it}^{\min} \le \theta_{it}^{\mathrm{in}} \le \theta_{it}^{\max}, \tag{49}$$

where $\theta_{it}^{\min}$ and $\theta_{it}^{\max}$ denote minimal and maximal temperature limits, respectively.

(4) Storage energy limits: The amount of cooling energy stored at any time $t \in \mathcal{T}$ should within an energy storage limit (capacity) $E_{\max}^{\mathrm{stored}}$, i.e.,

$$E_t^{\mathrm{stored}} \in [0, E_{\max}^{\mathrm{stored}}]. \tag{50}$$

(5) Energy exchange limits: The energy exchanged with the storage network for each building i at t needs to satisfy the following constraint:

$$-E_{i,\max}^{\mathrm{ch}} \le E_{it}^{\mathrm{ch}} \le E_{i,\max}^{\mathrm{ch}}, \tag{51}$$

where $E_{i,\max}^{\text{ch}} \in \mathbb{R}$ is the maximal energy that can be exchanged with the storage network for building i.

Define vectors $u_i = (\theta_{it}^{\text{in}}; t \in \mathcal{T})$ and $x = [E_1^{\text{ch}}; E_2^{\text{ch}}; \cdots; E_N^{\text{ch}}]$ with $E_i^{\text{ch}} = (E_{it}^{\text{ch}}; t \in \mathcal{T})$. Then u_i can be viewed as a local decision vector related to individual comfort, and x is a global decision vector related to the energy exchange of buildings with the sharing storage network. All constraints above can be treated as local constraints. Therefore, the energy management problem (46) could be viewed as an instance of Problem 1 such that our proposed algorithm can be applied to solve it in a distributed way.

2) Simulation results: In our simulation, we consider the cooling system shown in Fig. 3. The edges of the time-varying communication network are divided into two groups, i.e., the red and blue ones, which are activated alternatively with link weights equal to 1/2. This setting satisfies Assumption 3.1 with a period of $T = 2$. The parameters of biquadratic approximations and electricity limits of the chillers are taken from [44]. The energy coordination interval is from 20:00 P.M. on one day to 20:00 P.M., and the length of each interval is 1 h. The indoor temperature constraints are set to $\theta^{\max} = 25°\text{C}$ for buildings 1, 2, 3 and $\theta^{\max} = 26°\text{C}$ for buildings 4, 5, 6 during working hours (8:00 A.M. to 17:00 P.M.), and $\theta^{\max} = 27°\text{C}$ for all buildings at other times. We assume that $\theta^{\min} = 24°\text{C}$ for all buildings at all periods and indoor initial temperatures are all set to be 26°C. The capacity of the storage unit is $E_{\max}^{\text{stored}} = 15$ (kWh) and the maximal exchange energy $E_{i,\max}^{\text{ch}} = 2$ (kW) for all $i = 1, \cdots, N$. At the initial time, the cooling energy stored in this unit is assumed to be 0. The constant values c_i^{air} and R_i in equation (41) are taken from [42] with $\hat{\tau} = 1$ h. Fig. 4 shows hourly electricity prices and the outdoor temperature data for the considered summer day.

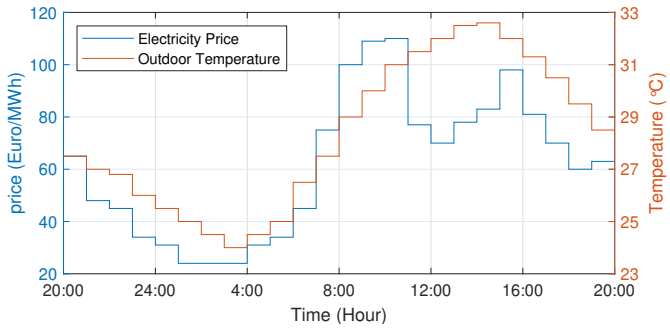


Fig. 4: Hourly energy prices and the outdoor temperature data for one summer day.

We apply Algorithm 1 to solve the underlying optimization problem over the time-varying communication network for the optimal solution. Fig. 5 displays the optimal indoor temperature profiles for the buildings returned by Algorithm 1 upon convergence. As we can see, all the temperature values are within the feasible range. In order to reduce the total system cost, it is preferable for each building to use less energy to meet their cooling demand. Therefore, the acceptable temperature profiles during working hours are the maximal comfort temperature limits $\theta^{\max}$. At other times, the user's tolerance for temperature is much higher with the limit of 27°C.

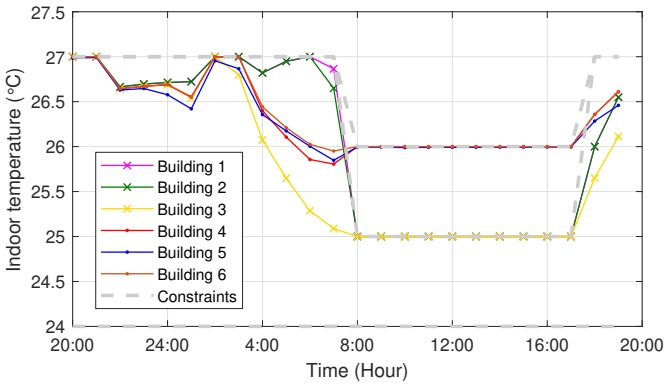


Fig. 5: The indoor buildings' temperatures returned by Algorithm 1.

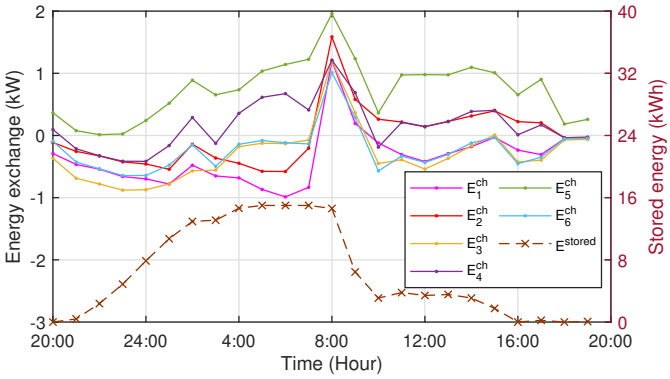


Fig. 6: The stored energy and the energy exchange of the buildings with the storage network.

Fig. 6 shows the stored energy and the energy exchange profiles of buildings computed by building 1 when Algorithm 1 converges. The profiles computed by other buildings are the same as building 1 due to the proposed consensus mechanism, hence we do not show them in Fig. 6 for simplicity. It can be observed that most buildings at night discharge ($E_{it}^{\text{ch}} < 0$) to the storage network due to the lower electricity prices shown in Fig. 4 and less cooling demand. From 8:00 A.M., as the demand increases, all buildings start to draw energy from the storage network. By the energy coordination, the differences in chiller sizes can be compensated through the sharing storage network, and the total system cost could also be minimized.

VI. Conclusion

In this paper, we proposed a novel fully-distributed heavy-ball algorithm for a class of consensus optimization problems with non-smooth objective functions and heterogenous constraints per agent over a time-varying communication network where each agent could only interact with its neighbours. In this algorithm, we combined a momentum term with the gradient tracking technique to accelerate its convergence. Under certain assumptions on the connectivity of the network and the agent weights, we proved that the estimates of each agent could converge to a common limit point, i.e., the global

optimal solution with proper step-size and momentum parameters. We also showed a convergence rate of $\mathcal{O}(\frac{\ln k}{\sqrt{k}})$ in objective value for a particular step-size choice. The performance of the proposed algorithm was illustrated by the simulation results on an example involving ℓ_1-regularized logistic regression, quantifying also numerically the improvement in the convergence rate relative to other distributed algorithms. The case study of energy management for building district cooling systems further demonstrates its efficacy.

The algorithm studied in this paper provides advanced guidance to the distributed application of accelerated momentum methods, especially for non-smooth constrained optimization problems. Future work concentrates on removing the double stochasticity assumptions and on considering a directed communication network.

APPENDIX

Proof of Lemma 4.5

In Algorithm 1, the projection operation of $x_i(k+1)$ is equivalent to

$$x_i(k+1) = \arg\min_{x\in\mathcal{X}_i} \Big\{ s_i(k)^\top x + \frac{1}{2\alpha(k)} \big\| x - z_i(k) - \beta(x_i(k) - x_i(k-1)) \big\|_2^2 \Big\}. \tag{52}$$

By optimality of $x_i(k+1)$, we have

$$\begin{aligned} & s_i(k)^\top x_i(k+1) + \frac{1}{\alpha(k)} \big(x_i(k+1) - z_i(k) \\ & \qquad - \beta(x_i(k) - x_i(k-1)\big)^\top x_i(k+1) \\ \le\ & s_i(k)^\top x^* + \frac{1}{\alpha(k)} \big(x_i(k+1) - z_i(k) \\ & \qquad - \beta(x_i(k) - x_i(k-1)\big)^\top x^*, \end{aligned} \tag{53}$$

where $s_i(k) + \frac{1}{\alpha(k)}\big(x_i(k+1) - z_i(k) - \beta(x_i(k) - x_i(k-1)\big)$ is the gradient of the objective function in (52), evaluated at $x_i(k+1)$. Then consider the following equality

$$\begin{aligned} & \frac{1}{\alpha(k)} \left(x_i(k+1) - z_i(k)\right)^\top \left(x_i(k+1) - x^*\right) \\ =\ & \frac{1}{2\alpha(k)} \left\|x_i(k+1) - z_i(k)\right\|_2^2 + \frac{1}{2\alpha(k)} \left\|x_i(k+1) - x^*\right\|_2^2 \\ & - \frac{1}{2\alpha(k)} \left\|z_i(k) - x^*\right\|_2^2. \end{aligned} \tag{54}$$

Combining (53) and (54) gives

$$\begin{aligned} & s_i(k)^\top x_i(k+1) + \frac{1}{2\alpha(k)} \left\|x_i(k+1) - z_i(k)\right\|_2^2 \\ & + \frac{1}{2\alpha(k)} \left\|x_i(k+1) - x^*\right\|_2^2 \\ \le\ & s_i(k)^\top x^* + \frac{1}{2\alpha(k)} \sum_{j=1}^N A_{ij}(k) \left\|x_j(k) - x^*\right\|_2^2 \\ & + \frac{\beta}{\alpha(k)} (x_i(k) - x_i(k-1))^\top (x_i(k+1) - x^*), \end{aligned} \tag{55}$$

that considers $\|z_i(k) - x^*\|_2^2 \le \sum_{j=1}^N A_{ij}(k) \|x_j(k) - x^*\|_2^2$ due to the convexity of $\|\cdot\|_2^2$.

By multiplying both sides of (55) by $2\alpha(k)$ and summing it for all $i \in \mathcal{N}$, it can be obtained that:

$$\begin{aligned} & 2\alpha(k) \sum_{i=1}^N s_i(k)^\top x_i(k+1) + \sum_{i=1}^N \|x_i(k+1) - z_i(k)\|_2^2 \\ & + \sum_{i=1}^N \|x_i(k+1) - x^*\|_2^2 \\ \le\ & 2\alpha(k) \sum_{i=1}^N s_i(k)^\top x^* + \sum_{i=1}^N \|x_i(k) - x^*\|_2^2 \\ & + 2\beta \sum_{i=1}^N (x_i(k) - x_i(k-1))^\top (x_i(k+1) - x^*) \end{aligned} \tag{56}$$

where the equality $\sum_{i=1}^N \sum_{j=1}^N A_{ij}(k) \|x_j(k) - x^*\|_2^2 = \sum_{i=1}^N \|x_i(k) - x^*\|_2^2$ holds due to the double stochasticity of $A(k)$.

Consider $e_i(k+1) \equiv x_i(k+1) - z_i(k)$ and the following relations:

$$\begin{aligned} & 2(x_i(k) - x_i(k-1))^\top (x_i(k+1) - x^*) \\ =\ & \|x_i(k+1) - x_i(k-1)\|_2^2 - \|x_i(k+1) - x_i(k)\|_2^2 \\ & + \|x_i(k) - x^*\|_2^2 - \|x_i(k-1) - x^*\|_2^2 \\ \le\ & \|x_i(k+1) - x_i(k)\|_2^2 + 2\|x_i(k) - x_i(k-1)\|_2^2 \\ & + \|x_i(k) - x^*\|_2^2 - \|x_i(k-1) - x^*\|_2^2, \end{aligned} \tag{57}$$

where the inequality holds by adding and subtracting $x_i(k)$ in the first term of right-hand side and applying the relation $\|a+b\|_2^2 \le 2\|a\|^2 + 2\|b\|_2^2$.

Substituting (57) into (56) gives:

$$\begin{aligned} & 2\alpha(k) \sum_{i=1}^N s_i(k)^\top \left(x_i(k+1) - x^*\right) + \sum_{i=1}^N \|e_i(k+1)\|_2^2 \\ & + \sum_{i=1}^N \left[\|x_i(k+1) - x^*\|_2^2 - \beta\|x_i(k) - x^*\|_2^2 \right] \\ \le\ & \sum_{i=1}^N \left[\|x_i(k) - x^*\|_2^2 - \beta\|x_i(k-1) - x^*\|_2^2 \right] \\ & + \beta \sum_{i=1}^N \|x_i(k+1) - x_i(k)\|_2^2 \\ & + 2\beta \sum_{i=1}^N \|x_i(k) - x_i(k-1)\|_2^2. \end{aligned} \tag{58}$$

Consider the first term of (58). By adding and subtracting

$\bar{v}(k+1)$, we have

$$\begin{aligned}
&2\alpha(k)\sum_{i=1}^{N} s_i(k)^\top \left(x_i(k+1)-x^*\right)\\
&=2\alpha(k)\sum_{i=1}^{N} s_i(k)^\top \left(x_i(k+1)-\bar{v}(k+1)\right)\\
&\quad+2\alpha(k)\sum_{i=1}^{N} s_i(k)^\top \left(\bar{v}(k+1)-x^*\right) \qquad (59)\\
&\geq -2\widehat{L}\alpha(k)\sum_{i=1}^{N}\|x_i(k+1)-\bar{v}(k+1)\|_2\\
&\quad+2\alpha(k)\sum_{i=1}^{N} s_i(k)^\top \left(\bar{v}(k+1)-x^*\right)
\end{aligned}$$

where the inequality holds by the Cauchy-Schwartz inequality and Lemma 4.2. In terms of the last term on the right-hand side of above (59), we have

$$\begin{aligned}
&\sum_{i=1}^{N} s_i(k)^\top \left(\bar{v}(k+1)-x^*\right)\\
&=\sum_{i=1}^{N} g_i(k)^\top \big(\bar{v}(k+1)-x_i(k+1)+x_i(k+1)\\
&\qquad\qquad -x_i(k)+x_i(k)-x^*\big) \qquad (60)\\
&\geq -L\sum_{i=1}^{N}\|\bar{v}(k+1)-x_i(k+1)\|_2\\
&\quad -L\sum_{i=1}^{N}\|x_i(k+1)-x_i(k)\|_2+\sum_{i=1}^{N} g_i(k)^\top \left(x_i(k)-x^*\right)
\end{aligned}$$

where the equality holds by using Lemma 4.1 and adding and subtracting $x_i(k)$ and $x_i(k+1)$ for all $i\in\mathcal{N}$, and the inequality follows from the Cauchy-Schwartz inequality and item (ii) of Assumption 2.1.

Consider the last term on the right-hand side of (60), and based on the subgradient property for a convex function $(g_i(k)\in\partial f_i(x_i(k)))$ we obtain that:

$$\begin{aligned}
&\sum_{i=1}^{N} g_i(k)^\top \left(x_i(k)-x^*\right)\\
&\geq \sum_{i=1}^{N}\big(f_i(x_i(k))-f_i(x^*)\big) \qquad (61)\\
&=\sum_{i=1}^{N}\big(f_i(x_i(k))-f_i(\bar{v}(k))\big)+\sum_{i=1}^{N}\big(f_i(\bar{v}(k))-f_i(x^*)\big)\\
&\geq -L\sum_{i=1}^{N}\|x_i(k)-\bar{v}(k)\|_2+\sum_{i=1}^{N}\big(f_i(\bar{v}(k))-f_i(x^*)\big),
\end{aligned}$$

where the equality follows by adding and subtracting $f_i(\bar{v}(k))$. The last inequality holds since the function f_i is Lipschitz continuous over $\mathcal{X}_i$ such that $|f_i(x)-f_i(y)|\leq L\|x-y\|_2, \forall x,y\in\mathcal{X}_i$ under Assumption 2.1, where constant L is defined in (2) [40].

Substituting (59), (60) and (61) into (58) gives

$$\begin{aligned}
&-2(L+\widehat{L})\alpha(k)\sum_{i=1}^{N}\|x_i(k+1)-\bar{v}(k+1)\|_2\\
&-2L\alpha(k)\sum_{i=1}^{N}\|x_i(k)-\bar{v}(k)\|_2-2L\alpha(k)\sum_{i=1}^{N}\|x_i(k+1)-x_i(k)\|_2\\
&-\beta\sum_{i=1}^{N}\|x_i(k+1)-x_i(k)\|_2^2-2\beta\sum_{i=1}^{N}\|x_i(k)-x_i(k-1)\|_2^2\\
&+2\alpha(k)\sum_{i=1}^{N}\big(f_i(\bar{v}(k))-f_i(x^*)\big)+\sum_{i=1}^{N}\|e_i(k+1)\|_2^2\\
&+\sum_{i=1}^{N}\left\|x_i(k+1)-x^*\right\|_2^2-\beta\|x_i(k)-x^*\|_2^2\\
&\leq\sum_{i=1}^{N}\left\|x_i(k)-x^*\right\|_2^2-\beta\|x_i(k-1)-x^*\|_2^2. \qquad (62)
\end{aligned}$$

In terms of the third term on the left-hand side of (62), by adding and subtracting $\bar{v}(k)$ and $x_i(k+1)=z_i(k)+e_i(k+1)$, we obtain

$$\begin{aligned}
&2L\alpha(k)\sum_{i=1}^{N}\|x_i(k+1)-x_i(k)\|_2\\
&=2L\alpha(k)\sum_{i=1}^{N}\|z_i(k)-\bar{v}(k)+e_i(k+1)+\bar{v}(k)-x_i(k)\|_2\\
&\leq 2L\alpha(k)\sum_{i=1}^{N}\sum_{j=1}^{N}A_{ij}(k)\|x_j(k)-\bar{v}(k)\|_2 \qquad (63)\\
&\quad+2L\alpha(k)\sum_{i=1}^{N}\|e_i(k+1)\|_2+2L\alpha(k)\sum_{i=1}^{N}\|\bar{v}(k)-x_i(k)\|_2\\
&=4L\alpha(k)\sum_{i=1}^{N}\|x_i(k)-\bar{v}(k)\|_2+2L\alpha(k)\sum_{i=1}^{N}\|e_i(k+1)\|_2\\
&\leq 4L\alpha(k)\sum_{i=1}^{N}\|x_i(k)-\bar{v}(k)\|_2+\gamma\sum_{i=1}^{N}\|e_i(k+1)\|_2^2+\frac{NL^2}{\gamma}\alpha(k)^2
\end{aligned}$$

where the first inequality follows from the triangle inequality, the second equality holds by exchanging the order of summation and using the double stochasticity of $A(k)$, and the last inequality holds due to $2xy\leq x^2+y^2$ with $x=\frac{L}{\sqrt{\gamma}}\alpha(k)$ and $y=\sqrt{\gamma}\|e_i(k+1)\|_2$ for some $\gamma\in(0,1)$. Similarly, regarding the fourth term on the left-hand side of (62), we have

$$\begin{aligned}
&\beta\sum_{i=1}^{N}\|x_i(k+1)-x_i(k)\|_2^2\\
&=\beta\sum_{i=1}^{N}\|z_i(k)-\bar{v}(k)+e_i(k+1)+\bar{v}(k)-x_i(k)\|_2^2\\
&\leq 3\beta\sum_{i=1}^{N}\|z_i(k)-\bar{v}(k)\|_2^2 \qquad (64)\\
&\quad+3\beta\sum_{i=1}^{N}\|e_i(k+1)\|_2^2+3\beta\sum_{i=1}^{N}\|\bar{v}(k)-x_i(k)\|_2^2\\
&\leq 3\beta\sum_{i=1}^{N}\|e_i(k+1)\|_2^2+6\beta\sum_{i=1}^{N}\|x_i(k)-\bar{v}(k)\|_2^2
\end{aligned}$$

where the first inequality holds due to $\|a+b+c\|_2^2 \leq 3\|a\|_2^2 + 3\|b\|_2^2 + 3\|c\|_2^2$, and the second inequality holds by applying the double stochasticity of $A(k)$ and convexity of $\|\cdot\|_2^2$.

Therefore, (27) in Lemma 4.5 could be obtained by substituting (63) and (64) into (62), thus concluding the proof.

REFERENCES

[1] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[2] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[3] B. Baingana, G. Mateos, and G. B. Giannakis, "Proximal-gradient algorithms for tracking cascades over social networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 4, pp. 563–575, 2014.

[4] B. Cai and Y. Zhang, "Different-level redundancy-resolution and its equivalent relationship analysis for robot manipulators using gradient-descent and zhang's neural-dynamic methods," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3146–3155, 2011.

[5] W. Ananduta, C. Ocampo-Martinez, and A. Nedić, "A distributed augmented lagrangian method over stochastic networks for economic dispatch of large-scale energy systems," *IEEE Transactions on Sustainable Energy*, vol. 12, no. 4, pp. 1927–1934, 2021.

[6] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–5.

[7] E. Ghadimi, H. R. Feyzmahdavian, and M. Johansson, "Global convergence of the heavy-ball method for convex optimization," in *European control conference*, 2015, pp. 310–315.

[8] T. Yang, Q. Lin, and Z. Li, "Unified convergence analysis of stochastic momentum methods for convex and non-convex optimization," *arXiv preprint arXiv:1604.03257*, 2016.

[9] W. Tao, G. W. Wu, and Q. Tao, "Momentum Acceleration in the Individual Convergence of Nonsmooth Convex Optimization With Constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 1107 – 1118, 2022.

[10] R. Xin and U. A. Khan, "Distributed Heavy-Ball: A Generalization and Acceleration of First-Order Methods with Gradient Tracking," *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2627–2633, 2020.

[11] H. Li, H. Cheng, Z. Wang, and G. C. Wu, "Distributed Nesterov Gradient and Heavy-Ball Double Accelerated Asynchronous Optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5723–5737, 2021.

[12] J. Gao, X. Liu, Y.-H. Dai, Y. Huang, and P. Yang, "A Family of Distributed Momentum Methods over Directed Graphs with Linear Convergence," *IEEE Transactions on Automatic Control*, pp. 1–8, 2022, doi: 10.1109/TAC.2022.3160684.

[13] J. Hu, Y. Yan, H. Li, Z. Wang, D. Xia, and J. Guo, "Convergence of an accelerated distributed optimisation algorithm over time-varying directed networks," *IET Control Theory and Applications*, vol. 15, no. 1, pp. 24–39, 2021.

[14] Y. Lü, H. Xiong, H. Zhou, and X. Guan, "A Distributed Optimization Accelerated Algorithm with Uncoordinated Time-Varying Step-Sizes in an Undirected Network," *Mathematics*, vol. 10, no. 3, pp. 1–17, 2022.

[15] P. Xie, K. You, R. Tempo, S. Song, C. Wu, C. Gu, Z. Wu, J. Li, and Y. Guo, "Distributed convex optimization with coupling constraints over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4331–4337, 2018.

[16] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592–606, 2012.

[17] C. Liu, H. Li, and Y. Shi, "A unitary distributed subgradient method for multi-agent optimization with different coupling sources," *Automatica*, vol. 114, pp. 1–12, 2020.

[18] S. Lee and A. Nedic, "Distributed random projection algorithm for convex optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 221–229, 2013.

[19] Q. Liu, S. Yang, and Y. Hong, "Constrained Consensus Algorithms with Fixed Step Size for Distributed Convex Optimization over Multiagent Networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 4259–4265, 2017.

[20] V. S. Mai and E. H. Abed, "Distributed optimization over directed graphs with row stochasticity and constraint regularity," *Automatica*, vol. 102, pp. 94–104, 2019.

[21] A. Falsone and M. Prandini, "Distributed decision-coupled constrained optimization via Proximal-Tracking," *Automatica*, vol. 135, pp. 1–12, 2022.

[22] W. Yu, H. Liu, W. X. Zheng, and Y. Zhu, "Distributed discrete-time convex optimization with nonidentical local constraints over time-varying unbalanced directed graphs," *Automatica*, vol. 134, pp. 1–15, 2021.

[23] X. Wu and J. Lu, "Fenchel dual gradient methods for distributed convex optimization over time-varying networks," *IEEE Transactions on Automatic Control*, vol. 64, no. 11, pp. 4629–4636, 2019.

[24] M. Zhu and S. Martinez, "On distributed convex optimization under inequality and equality constraints," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2012.

[25] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[26] P. Lin, W. Ren, and Y. Song, "Distributed multi-agent optimization subject to nonidentical constraints and communication delays," *Automatica*, vol. 65, pp. 120–131, 2016.

[27] K. Margellos, A. Falsone, S. Garatti, and M. Prandini, "Distributed constrained optimization and consensus in uncertain networks via proximal minimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1372–1387, 2018.

[28] L. Romao, K. Margellos, G. Notarstefano, and A. Papachristodoulou, "Subgradient averaging for multi-agent optimisation with different constraint sets," *Automatica*, vol. 131, pp. 1–14, 2021.

[29] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson, "Subgradient methods and consensus algorithms for solving convex optimization problems," in *47th IEEE Conference on Decision and Control*, 2008, pp. 4185–4190.

[30] S. Liang, L. Wang, and G. Yin, "Distributed quasi-monotone subgradient algorithm for nonsmooth convex optimization over directed graphs," *Automatica*, vol. 101, pp. 175–181, 2019.

[31] W. Shi, Q. Ling, G. Wu, and W. Yin, "A Proximal Gradient Algorithm for Decentralized Composite Optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 22, pp. 6013–6023, 2015.

[32] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[33] M. Gürbüzbalaban, A. Ozdaglar, and P. A. Parrilo, "On the convergence rate of incremental aggregated gradient algorithms," *SIAM Journal on Optimization*, vol. 27, no. 2, pp. 1035–1048, 2017.

[34] B. Polyak and P. Shcherbakov, "Lyapunov Functions: An Optimization Theory Perspective," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7456–7461, 2017.

[35] A. Falsone, I. Notarnicola, G. Notarstefano, and M. Prandini, "Tracking-ADMM for distributed constraint-coupled optimization," *Automatica*, vol. 117, p. 108962, 2020.

[36] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[37] A. Nedić, A. Olshevsky, and W. Shi, "Achieving Geometric Convergence for Distributed Optimization Over Time-Varying Graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.

[38] A. Falsone, K. Margellos, S. Garatti, and M. Prandini, "Dual decomposition for multi-agent distributed optimization with coupling constraints," *Automatica*, vol. 84, pp. 149–158, 2017.

[39] D. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[40] D. P. Bertsekas, *Convex Optimization Theory*. Athena Scientific, 2009.

[41] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[42] C. Zhang, Y. Xu, Z. Li, and Z. Y. Dong, "Robustly Coordinated Operation of a Multi-Energy Microgrid with Flexible Electric and Thermal Loads," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2765–2775, 2019.

[43] N. Li, L. Chen, and S. H. Low, "Optimal demand response based on utility maximization in power networks," in *IEEE Power and Energy Society General Meeting*, 2011.

[44] F. Belluschi, A. Falsone, D. Ioli, K. Margellos, S. Garatti, and M. Prandini, "Distributed optimization for structured programs and its application to energy management in a building district," *Journal of Process Control*, vol. 89, pp. 11–21, 2020.