Machine Intelligence Research
www.mi-research.net

19(1), February 2022, 75-88
DOI: 10.1007/s11633-022-1319-2

Continuous-time Distributed Heavy-ball Algorithm for Distributed Convex Optimization over Undirected and Directed Graphs

Hao-Ran Yang Wei Ni

School of Science, Nanchang University, Nanchang 330031, China

Abstract: This paper proposes second-order distributed algorithms over multi-agent networks to solve the convex optimization problem by utilizing the gradient tracking strategy, with convergence acceleration being achieved. Both the undirected and unbalanced directed graphs are considered, extending existing algorithms that primarily focus on undirected or balanced directed graphs. Our algorithms also have the advantage of abandoning the diminishing step-size strategy so that slow convergence can be avoided. Furthermore, the exact convergence to the optimal solution can be realized even under the constant step size adopted in this paper. Finally, two numerical examples are presented to show the convergence performance of our algorithms.

Keywords: Distributed convex optimization, second-order distributed algorithm, multi-agent systems, gradient tracking, directed graph.

Citation: H. R. Yang, W. Ni. Continuous-time distributed heavy-ball algorithm for distributed convex optimization over undirected and directed graphs. *Machine Intelligence Research*, vol.19, no.1, pp.75–88, 2022. http://doi.org/10.1007/s11633-022-1319-2

1 Introduction

In past decades, due to rapid developments of big data and parallel computing, distributed optimization has become a core modeling framework of many large-scale problems and found numerous applications in formation control[1, 2], resource allocation[3], path planning[4], state consensus problem[5], power control[6], sensor networks[7], etc. Compared to traditional optimization methods, the distributed optimization strategy divides complex optimization tasks into many simple ones by different agents in a network, and allows information communications among these agents so that they can compute the optimal solution to the original optimization problem in a cooperative and consensus way (see [8–16] for more details).

For distributed optimization, various algorithms have been developed, with the gradient-based algorithm being the most popular; typical examples include the distributed gradient descent algorithm[9, 10] and the distributed dual averaging gradient algorithm[11, 12]. In addition to these simple forms, more complex gradient-based algorithms have been developed, such as the push-sum distributed algorithm[13], which is based on dual averaging scheme, the surplus-based distributed gradient projection algorithm[14], fast distributed gradient algorithm[15], the distributed gossip-based algorithm with low communication cost[16], etc. It is worth pointing out that the strategy of diminishing step sizes is usually adopted in the aforementioned algorithms to obtain the optimal solution of the optimization problem. Unfortunately, despite its popularity, it has been reported that the diminishing step-size scheme suffers a slow convergence rate[8, 17] and thus faces limited applications.

With constant step size, the convergence rate of the distributed gradient-based algorithm can be improved[8, 18]. However, algorithms with constant step size converge only to a small neighborhood of the optimal solution due to the use of the local gradient in each agent; namely, an inexact optimal solution is obtained (see [10, 18, 19] for details). To reconcile slow convergence and an inexact solution, the gradient tracking strategy is merged with the distributed convex optimization algorithms where an estimate of the global average gradients is used to replace the local gradient in each agent[20–23]. Using the gradient tracking strategy, the augment distributed gradient algorithm[20], the push-pull gradient algorithm[21], the adapt-then-combine distributed inexact gradient tracking algorithm[22], the projection-free algorithm[23], and the distributed Nash equilibrium seeking algorithm[24] can achieve an exact optimal solution and improve the con-

Research Article
Manuscript received July 19, 2021; accepted November 18, 2021
Recommended by Associate Editor Long Cheng
Colored figures are available in the online version at https://link.springer.com/journal/11633

vergence rate under constant step size. However, the time-derivatives of the gradients or the agents' state are required for most existing gradient tracking algorithms when writing them in continuous-time forms. For example, for Algorithm 1 in [23], the term $\nabla f_i(x_i(k+1))-\nabla f_i(x_i(k))$ in discrete-time becomes $\dot{\nabla} f_i(x_i(t))$ in continuous-time; for the equation (4) in [24], the term $x_i(k+1)-x_i(k)$ in discrete-time becomes $\dot{x}_i(t)$ in continuous-time. Aside from the fact that the time-derivative of the gradient or the agent′ state is hard to obtain, this kind of derivative-dependent information is harmful in algorithm design because it is sensitive to noise. To avoid the use of derivative-dependent information in gradient tracking design, we use the proportional-integral (PI) consensus dynamic[25] to achieve the gradient tracking strategy in our continuous-time distributed algorithm under constant step size. Our algorithm is different from those in [20–24] since there is no requirement for the time-derivatives of the gradients or the agents' state in our continuous-time algorithms. While similar PI-based gradient tracking is also applied to the distributed algorithm design in [26], the algorithm in [26] uses global information − the number N of the agents; please refer to the equation (8a) in [26]. Similar global information also appears in [27] (see the equation (10) in [27]). In practice, this global information is generally hard to obtain in distributed algorithm design. Furthermore, in some circumstances, the network may be changing with the addition or deletion of the agents to the network, and consequently the algorithms presented in [26, 27] do not apply to account for this situation because N is not a constant. The gradient tracking algorithm proposed in our paper does not depend on this global information.

Compared with distributed first-order algorithms, distributed second-order algorithms can achieve significant convergence acceleration for twice continuously differentiable and strongly convex cost functions[28, 29], which attract considerations in the distributed algorithm design. For example, in [30], based on the event-triggered communication strategy, a distributed continuous-time Newton-Raphson algorithm with a fast convergence rate and low communication burden is developed for the convex optimization problem. Tran et al.[31] investigate the distributed algorithm for double-integrator systems with exogenous disturbances and propose a novel algorithm that utilizes the internal model principle to asymptotically compensate for the disturbance. Also, Deng et al.[32] present two second-order algorithms to deal with the resource allocation problem. Recall the fact that we have emphasized before that the technique of gradient tracking allows distributed optimization algorithms to compute the exact optimal solution even under constant step size. Despite the fast convergence rate of distributed second-order optimization algorithms, there are few considerations for employing the gradient tracking strategy to the second-order optimization algorithms. The above case has been considered in [33], where the average consensus protocol is employed to develop a discrete-time distributed second-order Newton-Raphson consensus algorithm for tracking the averages of Hessian and gradient. However, the tracking strategy is not easy to implement in the case of continuous-time. Most of the literature applies gradient tracking to the discrete-time first-order or second-order distributed optimization algorithms[20−23, 33]; our paper is an application of the gradient tracking method to the continuous-time second-order heavy-ball optimization algorithm[34] by resorting to the PI consensus dynamic[25].

Apart from the reviewed issues above, we point out that generalizing the results of existing algorithms from undirected graphs to directed graphs are also important considerations in the distributed convex optimization. The difficulty of this generalization lies in the fact that the symmetry of the Laplacian matrix is broken if the undirected graph is generalized to directed ones. Although some distributed second-order algorithms are developed for directed graphs[35, 36], these algorithms require the graph to be balanced. This assumption makes a trivial generalization from undirected graphs to directed ones since the Laplacian matrix of the balanced graph is still symmetric. Thus, the analysis techniques in [35, 36] are roughly the same as those of undirected graphs. A nontrivial extension appears in [14], where a distributed gradient projection algorithm with the surplus consensus protocol is developed for an unbalanced directed graph. However, the algorithm in [14] requires each agent to know its out-degree to structure a column stochastic matrix, which is impractical in many situations. To remove this disadvantage, Wang et al.[37, 38] propose some distributed algorithms for unbalanced directed graphs with appropriate row-stochastic matrices and thus eliminate the requirement for agents' out-degree. Utilizing the left eigenvector of the Laplacian matrix associated with the zero eigenvalues, the distributed algorithm is designed to tackle the general unbalanced directed graph in [39]. However, the algorithm in [39] fails to converge to the optimal solution when the left eigenvector is unknown in advance. Although a continuous-time first-order algorithm is developed for an unbalanced directed graph by estimating the left eigenvector in [40], the work does not consider the second-order algorithm. Considering that the similar continuous-time distributed heavy-ball algorithm[41] only deals with the relatively simple case of an undirected graph, we generalize our original second-order heavy-ball algorithm from an undirected graph to an unbalanced directed graph in this paper. To be specific, by designing an auxiliary variable to estimate the left eigenvector of the Laplacian matrix associated with the zero eigenvalues, we develop a continuous-time distributed second-order algorithm under constant step size to achieve a fast convergence to the optimal solution over an unbalanced directed communication graph.

In summary, we combine the gradient tracking strategy with the speed-up technique to develop a novel continuous-time distributed convex optimization algorithm for undirected graphs. Besides gradient tracking, our algorithm uses the second-order heavy-ball optimization algorithm and removes the use of diminishing step sizes. Also, it can be regarded as the second-order extension of the first-order distributed gradient descent algorithm, thereby speeding up the convergence. In addition, to maintain exact convergence under constant step size, we apply the PI consensus dynamic[25] to the second-order heavy-ball optimization algorithm[34]. Moreover, in terms of directed communication graphs, we adopt an auxiliary variable to estimate the left eigenvector of the Laplacian matrix associated with the zero eigenvalues, and further apply our original algorithm to the distributed convex optimization over unbalanced directed graphs. In short, the main contributions of this paper are as follows.

1) We propose a continuous-time distributed second-order convex optimization algorithm under undirected graphs by extending the classical distributed gradient descent algorithm[10], with the additional improvement of dropping the diminishing step sizes, realizing exact convergence even under constant step size, and speeding up convergence to the optimal solution.

2) For our proposed second-order heavy-ball algorithm under undirected graphs, we adopt the idea of using average gradients in distributed optimization algorithm design to meet the optimality condition under constant step size and use the strategy of gradient tracking to make our algorithms distributed. Moreover, we adopt the PI consensus dynamic[25] to achieve gradient tracking in the continuous-time second-order heavy-ball algorithm, which gains some advantages over those tracking strategies in the first-order algorithms[20−24]; namely that we do not utilize the undesirable information such as the time-derivatives of the gradients or the agents′ state. While the global information N is required in the existing PI-based gradient tracking algorithm[26], our proposed algorithm removes the dependence of this global information. Also, compared with the second-order algorithm in [33], our proposed second-order algorithm does not require the estimation of the Hessian matrix.

3) We also consider the case of an unbalanced directed communication graph in the distributed convex optimization. In contrast to the application of gradient tracking and the heavy-ball algorithm under undirected graphs[20, 33, 41], we generalize our distributed second-order heavy-ball algorithm with gradient tracking from an undirected graph to an unbalanced directed graph through the estimation scheme of the left eigenvector of Laplacian matrix. Also, our algorithm overcomes the limitation of knowing the left eigenvector in [39].

The rest of this paper is arranged as follows. Section 2 includes some preliminaries and the problem formulations. A second-order distributed algorithm with gradient tracking strategy over undirected graph is proposed, and its convergence is analyzed in Section 3. Section 4 introduces a modified second-order distributed algorithm over an unbalanced directed graph and provides the corresponding convergence analysis. Two simulation examples are given in Section 5. A brief conclusion and future work are given in Section 6.

2 Preliminaries and problem formulations

2.1 Preliminaries

The set of real and positive real numbers is denoted by $\mathbf{R}$ and $\mathbf{R}_{>0}$, the set of n-dimensional column vectors is denoted by $\mathbf{R}^n$. We use 1_n and 0_n to represent the column vectors of n ones and zeros, respectively. I_n denotes the n-dimensional identity matrix. For vectors $x_1,\cdots,x_N\in\mathbf{R}^n$, we use the notation $x=\mathrm{col}(x_1,\cdots,x_N)=(x_1^{\mathrm{T}},\cdots,x_N^{\mathrm{T}})^{\mathrm{T}}$ to denote a new stacked vector. We use $\mathrm{diag}(A_1,\cdots,A_N)$ to represent the block diagonal matrix A whose diagonal elements are $A_1,\cdots,A_N$. We let $\|x\|=\sqrt{x^{\mathrm{T}}x}$ denote the standard Euclidean norm of a vector x. Also, the Kronecker product of arbitrary matrices $A\in\mathbf{R}^{m\times n}$ and $B\in\mathbf{R}^{p\times q}$ is defined as $A\otimes B\in\mathbf{R}^{mp\times nq}$. The positive definiteness of a symmetric matrix A is denoted as $A>0$. Besides, $\nabla f(\cdot)$ is the gradient of a function $f(\cdot)$.

Then we provide some basic definitions of graph theory[42]. We consider a graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$, where $\mathcal{V}=\{1,\cdots,N\}$ is the set of nodes and $\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$ is the set of edges. An edge $(i,j)\in\mathcal{E}$ means that node i can receive information from node j. A graph $\mathcal{G}$ is undirected if the edges $(i,j)\in\mathcal{E}$ and $(j,i)\in\mathcal{E}$ are the same. The set of neighbors of node i is denoted as $N_i=\{j\in\mathcal{V}:(i,j)\in\mathcal{E}\}$. A directed path is a sequence of nodes connected by edges. A directed graph is strongly connected if, for every pair of nodes, there is a directed path connecting them. The adjacency matrix of $\mathcal{G}$ is denoted by $A=[a_{ij}]\in\mathbf{R}^{N\times N}$, where $a_{ij}>0$ if $(i,j)\in\mathcal{E}$ and $a_{ij}=0$ otherwise. The Laplacian matrix of $\mathcal{G}$ is $L=D-A$ with $D=\mathrm{diag}(d_1,\cdots,d_N)$, where $d_i=\sum_{j=1}^N a_{ij}$ for each $i\in\mathcal{V}$. Note that the Laplacian matrix L of the undirected graph is symmetric, positive semi-definite, and $L1_N=0_N$. The real eigenvalues of L are denoted by $\lambda_1,\cdots,\lambda_N$ with $\lambda_1\le\lambda_2\le\cdots\le\lambda_N$. If the undirected graph $\mathcal{G}$ is connected, L has a zero eigenvalue, and the rest of the eigenvalues are positive.

2.2 Problem formulation

We consider a network consisting of N agents, with the interaction described by a graph $\mathcal{G}$. Each agent $i\in\mathcal{V}$ has a local convex cost function $f_i:\mathbf{R}^n\to\mathbf{R}$, and the global cost function $f(x)$ in the network is the sum of these N local cost functions $f(x)=\sum_{i=1}^{N} f_i(x)$. The aim is to minimize the global cost function:

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

by designing a local optimization algorithm on each agent and by making use of information communications of each agent with its neighbor in the network so that the optimal solution can be cooperatively computed by these N agents in a consensus way. Optimization algorithms structured in this way are called distributed algorithms. In this paper, we generalize the existing first-order distributed convex algorithms to the second-order to achieve faster convergence rate and abandon some undesirable technical treatments in designing distributed optimization algorithms, such as diminishing step sizes and inexact convergence.

To facilitate the subsequent analysis, Assumptions 1−3 and Definition 1 are made.

Assumption 1 (Connectivity). The graph $\mathcal{G}$ is strongly connected.

Assumption 2 (Strong convexity). For each $i\in\mathcal{V}$, the local cost function f_i is twice continuously differentiable and s_i-strongly convex, i.e.,

$$(x-y)^{\mathrm{T}}[\nabla f_i(x)-\nabla f_i(y)]\geq s_i\|x-y\|^2, \quad \forall x,y\in\mathbf{R}^n$$

for parameter $s_i>0$.

Assumption 2 guarantees the existence and uniqueness of an optimal solution to the convex optimization problem (1), and it is widely used in [3, 32, 41].

Assumption 3 (Lipschitz continuous). For each $i\in\mathcal{V}$, the gradient of local cost function f_i is ℓ_i-Lipschitz continuous, i.e.,

$$\|\nabla f_i(x)-\nabla f_i(y)\|\leq \ell_i\|x-y\|, \quad \forall x,y\in\mathbf{R}^n$$

for parameter $\ell_i>0$.

Definition 1 (Strong monotonicity). A mapping $x\to f:\mathbf{R}^n\to\mathbf{R}$ is μ-strongly monotone if

$$(x-y)^{\mathrm{T}}[\nabla f(x)-\nabla f(y)]\geq \mu\|x-y\|^2, \quad \forall x,y\in\mathbf{R}^n$$

for some $\mu\in\mathbf{R}_{>0}$.

Assumptions 1−3 are standard in the literature of distributed convex optimization, see [8, 32, 43] for example.

If Assumption 2 is satisfied, then the convex optimization problem (1) has a unique optimal solution $x^*=\arg\min_{x\in\mathbf{R}^n} f(x)$ satisfying the optimality condition (see [44]):

$$\sum_{i=1}^{N}\nabla f_i(x^*)=0_n. \tag{2}$$

3 Distributed convex optimization over undirected graph

In this section, a continuous-time distributed convex optimization algorithm of second-order is presented to solve the convex optimization problem (1) over an undirected graph. Also, the convergence result of the proposed algorithm is proved in detail.

3.1 Algorithm design

To solve the convex optimization problem (1) in a distributed way and to improve existing first-order algorithms, we design for each agent $i\in\mathcal{V}$ the following second-order accelerated algorithm.

$$\begin{cases}\dot{x}_i=v_i & \text{(3a)}\\ \dot{v}_i=-v_i-y_i+\sum_{j\in N_i}(x_j-x_i) & \text{(3b)}\\ \dot{y}_i=-y_i+k\nabla f_i(x_i)+\sum_{j\in N_i}(z_j-z_i) & \text{(3c)}\\ \dot{z}_i=-\sum_{j\in N_i}(y_j-y_i) & \text{(3d)}\end{cases}$$

where $x_i\in\mathbf{R}^n$ is a local estimation of the optimal solution x^* by agent i, $y_i\in\mathbf{R}^n$ is a local estimation of the average gradients $\dfrac{1}{N}\sum_{i=1}^{N}\nabla f_i(x_i)$ by agent i, $v_i,z_i\in\mathbf{R}^n$, and k is a positive parameter to be designed. Since each agent $i\in\mathcal{V}$ only has access to its own local cost function f_i but communicates only with its neighbors, algorithm (3) is guaranteed to be distributed.

Although distributed optimization algorithms of first-order type (for example, distributed gradient descent algorithm[9, 10], distributed dual averaging gradient algorithm[11, 12], and fast gradient algorithm[15]) are largely reported in the literature, accelerating the convergence of first-order algorithms is also an important aspect behind many of future research. In our algorithm, the equations (3a) and (3b) adopt the distributed form of the second-order heavy-ball algorithm[34, 45], which extends the first-order distributed gradient descent algorithm[10] and has the advantage of accelerated convergence rate. In order to highlight the difference of the algorithm (3) from existing second-order distributed optimization algorithms, a Remarks 1–3 are made as follows.

Remark 1. The distributed heavy-ball algorithm reported in literature usually takes the form:

$$\begin{cases}\dot{x}_i=v_i & \text{(4a)}\\ \dot{v}_i=-v_i-\alpha(t)\nabla f_i(x_i)+\sum_{j\in N_i}(x_j-x_i) & \text{(4b)}\end{cases}$$

using a diminishing step size $\alpha(t)$, which, however, degrades the acceleration performance of the convergence. This degradation phenomenon due to the diminishing step-size rule is also observed in the first-order algorithms such as the distributed gradient-push algorithm[13] and the distributed gossip-based algorithm[16], to name a few. One popular way to overcome slow convergence caused by the diminishing step sizes is to adopt a constant step size, with refined performances being reported in [8, 18]. While the constant step-size strategy has a faster convergence than that of the diminishing step-size scheme, the former only converges to a small neighborhood of the optimal solution x^* since the optimality condition (2) in the optimal solution x^* is not satisfied; see [18, 19] for example. In an endeavor to design a distributed second-order convex optimization retaining the merit of exact convergence and avoiding the degradation of the convergence rate, we propose a novel design strategy in this paper, which uses a constant step-size version of (3b) and modifies the local gradient $\nabla f_i(x_i)$ into the average of full gradients so that the optimality condition (2) can be ensured, and then employs a distributed gradient tracking algorithm (3c) and (3d) to give a local estimation of the average gradients such that the algorithm is distributed.

Remark 2. Although [33, 45] also consider the classical gradient tracking strategy in the second-order distributed convex optimization, these algorithms are discrete-time and depend on $\nabla f_i(x_i(k+1)) - \nabla f_i(x_i(k))$ whose continuous-time version is exactly the time-derivative of the gradient $\dot{\nabla} f_i(x_i(t))$. A similar issue also appears in [24], in which the gradient tracking relies on utilizing the state information $x_i(k+1) - x_i(k)$ whose continuous-time version is nothing more than the time-derivative of the state $\dot{x}_i(t)$. However, this kind of derivative-dependent information cannot be easily accessed and is not robust to the perturbation of noise in the algorithm design. To remove this disadvantage of using such derivative-dependent information, we adopt in our optimization algorithm a totally different distributed gradient tracking approach from those in [24, 33, 45] with the help of PI dynamic average consensus[25].

Remark 3. Although both [26] and our algorithm (3) adopt similar PI dynamic consensus in the continuous-time distributed algorithm design, global information − the number N of the agents, is required in the algorithm [26] (see the equation (8a) in [26]). The algorithm proposed in [26] does not apply if agents are added or removed from the network, with a changing number of the agents to be rendered. Also, this kind of global information is not easy to access in distributed algorithm design. Consequently, our algorithm (3) removes the dependence of this global information.

To facilitate the subsequent analysis, algorithm (3) can be rewritten into the following compact form:

$$\begin{cases} \dot{x} = v \\ \dot{v} = -v - y - \mathcal{L}x \\ \dot{y} = -y + kG(x) - \mathcal{L}z \\ \dot{z} = \mathcal{L}y \end{cases} \tag{5}$$

where $\mathcal{L} \triangleq L \otimes I_n$, $x = \mathrm{col}(x_1, \cdots, x_N) \in \mathbf{R}^{nN}$, $v = \mathrm{col}(v_1, \cdots, v_N) \in \mathbf{R}^{nN}$, $y = \mathrm{col}(y_1, \cdots, y_N) \in \mathbf{R}^{nN}$, $z = \mathrm{col}(z_1, \cdots, z_N) \in \mathbf{R}^{nN}$ and $G(x) = \mathrm{col}(\nabla f_1(x_1), \cdots, \nabla f_N(x_N)) \in \mathbf{R}^{nN}$.

3.2 Convergence analysis

We first analyze the relationship between the equilibrium (x^*, v^*, y^*, z^*) of the dynamics (5) and the optimal solution x^* of the convex optimization problem (1). Also, the convergence analysis of the dynamics (3) with the equilibrium (x^*, v^*, y^*, z^*) is presented under certain conditions.

Lemma 1. For the convex optimization problem (1) under Assumptions 1–3, let (x^*, v^*, y^*, z^*) be an equilibrium of the dynamics (5). Then, $(x^*, v^*, y^*, z^*) = (1_N \otimes x^*, 0_{nN}, 0_{nN}, z^*)$ with z^* arbitrarily satisfies $\mathcal{L}z^* = kG(x^*)$.

Proof. Since (x^*, v^*, y^*, z^*) is the equilibrium of the dynamics (5), we substitute it into the dynamics (5) to obtain

$$\begin{cases} 0_{nN} = v^* & \text{(6a)} \\ 0_{nN} = -v^* - y^* - \mathcal{L}x^* & \text{(6b)} \\ 0_{nN} = y^* + kG(x^*) - \mathcal{L}z^* & \text{(6c)} \\ 0_{nN} = \mathcal{L}y^*. & \text{(6d)} \end{cases}$$

In view of Assumption 1 and from (6d), one has $y^* = 1_N \otimes c$ for some $c \in \mathbf{R}^n$. Inserting this and (6a) into (6b) gives $1_N \otimes c + \mathcal{L}x^* = 0_{nN}$. Left multiplying both sides of the resulting equation by $(1_N^{\mathrm{T}} \otimes I_n)$, we obtain $c = 0$ and consequently $y^* = 0_{nN}$. Therefore, (6b) becomes $\mathcal{L}x^* = 0$, which implies $x^* = 1_N \otimes a$ for some $a \in \mathbf{R}^n$. Also, from (6c), we obtain $\mathcal{L}z^* = kG(x^*)$. Similarly, left multiplying both sides of (6c) by $(1_N^{\mathrm{T}} \otimes I_n)$, one has $(1_N^{\mathrm{T}} \otimes I_n)G(x^*) = 0_n$, which shows that $\sum_{i=1}^{N} \nabla f_i(a) = 0_n$. On the other hand, under Assumption 2, there exists a unique optimal solution x^* of the convex optimization problem (1) satisfying the optimality condition (2). The uniqueness of the optimal solution x^* gives $a = x^*$. Noting that $x^* = 1_N \otimes a$, it shows $x^* = 1_N \otimes x^*$. Concluding the above analysis, one sees that $(x^*, v^*, y^*, z^*) = (1_N \otimes x^*, 0_{nN}, 0_{nN}, z^*)$ with $\mathcal{L}z^* = kG(x^*)$. □

The convergence of the dynamics (5) to the equilibrium is analyzed in Theorem 1.

Theorem 1. For the convex optimization problem (1) under Assumptions 1–3, let $s = \min(s_1, \cdots, s_N)$, $\ell = \max(\ell_1, \cdots, \ell_N)$, and assume $\lambda_2 > (2\lambda_N + 1)/(4s - 6\ell^2)$, $\lambda_2 + \lambda_N \leq 1$, so that there exists a parameter k such that

$$\frac{2\lambda_N + 1}{4s - 6\ell^2} < k < \lambda_2. \tag{7}$$

Then the trajectory of dynamics (3) with any initial condition $(x_i(0), v_i(0), y_i(0), z_i(0))$ achieve $x_i(t) \to x^*$ asymptotically as $t \to \infty$ for each $i \in \mathcal{V}$.

Proof. We first transfer the equilibrium (x^*, v^*, y^*, z^*) of the dynamics (5) into the origin by letting $\widetilde{x} = x - x^*$, $\widetilde{v} = v - v^*$, $\widetilde{y} = y - y^*$, $\widetilde{z} = z - z^*$. Now the dynamics (5) under the new coordinates can be described as

$$\begin{cases} \dot{\tilde{x}} = \tilde{v} \\ \dot{\tilde{v}} = -\tilde{v} - \tilde{y} - \mathcal{L}\tilde{x} \\ \dot{\tilde{y}} = -\tilde{y} + kh - \mathcal{L}\tilde{z} \\ \dot{\tilde{z}} = \mathcal{L}\tilde{y} \end{cases} \tag{8}$$

where $h \triangleq G(x) - G(x^*)$. Then, we proceed to divide the dynamics (8) into two tractable sub-dynamics. In view of Assumption 1, there exists an orthogonal matrix $Q = [p, P]$ such that

$$Q^{\mathrm{T}}LQ = \mathrm{diag}\{0, \lambda_2, \cdots, \lambda_N\} \triangleq \begin{pmatrix} 0 & 0 \\ 0 & L_\triangle \end{pmatrix}_{N\times N} \tag{9}$$

where $L_\triangle \triangleq P^{\mathrm{T}}LP$, $p = 1_N/\sqrt{N} \in \mathbf{R}^{N\times 1}$, $P \in \mathbf{R}^{N\times(N-1)}$, $P^{\mathrm{T}}P = I_{N-1}$ and $PP^{\mathrm{T}} = I_N - pp^{\mathrm{T}}$. With the orthogonal matrix Q, we can define the following new variables:

$$\begin{aligned} \hat{x} &= \mathrm{col}(\hat{x}_1, \hat{x}_{2N}) = [p\otimes I_n, P\otimes I_n]^{\mathrm{T}}\tilde{x} \\ \hat{v} &= \mathrm{col}(\hat{v}_1, \hat{v}_{2N}) = [p\otimes I_n, P\otimes I_n]^{\mathrm{T}}\tilde{v} \\ \hat{y} &= \mathrm{col}(\hat{y}_1, \hat{y}_{2N}) = [p\otimes I_n, P\otimes I_n]^{\mathrm{T}}\tilde{y} \\ \hat{z} &= \mathrm{col}(\hat{z}_1, \hat{z}_{2N}) = [p\otimes I_n, P\otimes I_n]^{\mathrm{T}}\tilde{z} \end{aligned} \tag{10}$$

where $\hat{x}_1, \hat{v}_1, \hat{y}_1, \hat{z}_1 \in \mathbf{R}^n$, $\hat{x}_{2N}, \hat{v}_{2N}, \hat{y}_{2N}, \hat{z}_{2N} \in \mathbf{R}^{n(N-1)}$. Then, the dynamics (8) can be further transformed into the following two sub-dynamics:

$$\begin{cases} \dot{\hat{x}}_1 = \hat{v}_1 \\ \dot{\hat{v}}_1 = -\hat{v}_1 - \hat{y}_1 \\ \dot{\hat{y}}_1 = -\hat{y}_1 + k(p^{\mathrm{T}}\otimes I_n)h \\ \dot{\hat{z}}_1 = 0_n \end{cases} \tag{11a}$$

$$\begin{cases} \dot{\hat{x}}_{2N} = \hat{v}_{2N} \\ \dot{\hat{v}}_{2N} = -\hat{v}_{2N} - \hat{y}_{2N} - \mathcal{L}_\triangle\hat{x}_{2N} \\ \dot{\hat{y}}_{2N} = -\hat{y}_{2N} + k(P^T\otimes I_n)h - \mathcal{L}_\triangle\hat{z}_{2N} \\ \dot{\hat{z}}_{2N} = \mathcal{L}_\triangle\hat{y}_{2N} \end{cases} \tag{11b}$$

where $\mathcal{L}_\triangle \triangleq (L_\triangle \otimes I_n)$. Thus, we only need to analyze the stability of the zero solution of sub-dynamics (11). Inspired by the Lyapunov function used in [32], we construct a Lyapunov candidate $V = V_1 + V_2$, with V_1 and V_2 given below:

$$\begin{aligned} V_1 =& \frac{1}{2}(\|\hat{x}_1 - \hat{y}_1\|^2 + \|\hat{x}_{2N} - \hat{y}_{2N}\|^2) + \frac{1}{2}\|\hat{z}_{2N} + \\ & \hat{y}_{2N}\|^2 + \frac{1}{2}\|\hat{y}_1\|^2 + \frac{1}{2}(\|\hat{v}_1\|^2 + \|\hat{v}_{2N}\|^2) \end{aligned} \tag{12a}$$

$$V_2 = \frac{1}{2}\hat{z}_{2N}^{\mathrm{T}}\hat{z}_{2N} + \frac{1}{2}\hat{z}_{2N}^{\mathrm{T}}(\mathcal{L}_\triangle)^{-1}\hat{z}_{2N} + \frac{1}{2}\hat{x}_{2N}^{\mathrm{T}}\mathcal{L}_\triangle\hat{x}_{2N}. \tag{12b}$$

Since $Q = [p, P]$ is an orthogonal matrix, it is obvious that $\hat{x}_1^{\mathrm{T}}(p^{\mathrm{T}}\otimes I_n) + \hat{x}_{2N}^{\mathrm{T}}(P^{\mathrm{T}}\otimes I_n) = \tilde{x}^{\mathrm{T}}$ and $\hat{y}_1^{\mathrm{T}}(p^{\mathrm{T}}\otimes I_n) + \hat{y}_{2N}^{\mathrm{T}}(P^{\mathrm{T}}\otimes I_n) = \tilde{y}^{\mathrm{T}}$. Also, by (10), we have $\hat{x}_1^{\mathrm{T}}\hat{v}_1 + \hat{x}_{2N}^{\mathrm{T}}\hat{v}_{2N} = \hat{x}^{\mathrm{T}}\hat{v}$, $\hat{x}_1^{\mathrm{T}}\hat{y}_1 + \hat{x}_{2N}^{\mathrm{T}}\hat{y}_{2N} = \hat{x}^{\mathrm{T}}\hat{y}$, $\hat{v}_1^{\mathrm{T}}\hat{y}_1 + \hat{v}_{2N}^{\mathrm{T}}\hat{y}_{2N} = \hat{v}^{\mathrm{T}}\hat{y}$ and $\hat{v}_1^{\mathrm{T}}\hat{v}_1 + \hat{v}_{2N}^{\mathrm{T}}\hat{v}_{2N} = \hat{v}^{\mathrm{T}}\hat{v}$. With these results, the time-derivatives of V_1 and V_2 along the trajectories of (11) can be calculated as

$$\begin{aligned} \dot{V}_1 =& -k\tilde{x}^{\mathrm{T}}h + 2k\tilde{y}^{\mathrm{T}}h + \hat{x}^{\mathrm{T}}(\hat{v}+\hat{y}) - \hat{v}^{\mathrm{T}}\hat{v} - 2\hat{v}^{\mathrm{T}}\hat{y} - \\ & 2\hat{y}^{\mathrm{T}}\hat{y} + k\hat{z}_{2N}^{\mathrm{T}}(P^{\mathrm{T}}\otimes I_n)h + (\hat{x}_{2N} - \hat{y}_{2N})^{\mathrm{T}}\mathcal{L}_\triangle\hat{z}_{2N} - \\ & \hat{z}_{2N}^{\mathrm{T}}\hat{y}_{2N} - \hat{v}_{2N}^{\mathrm{T}}\mathcal{L}_\triangle\hat{x}_{2N} + \hat{y}_{2N}^{\mathrm{T}}\mathcal{L}_\triangle\hat{y}_{2N} - \hat{z}_{2N}^{\mathrm{T}}\mathcal{L}_\triangle\hat{z}_{2N} \end{aligned} \tag{13a}$$

$$\dot{V}_2 = \hat{y}_{2N}^{\mathrm{T}}\mathcal{L}_\triangle\hat{z}_{2N} + \hat{z}_{2N}^{\mathrm{T}}\hat{y}_{2N} + \hat{x}_{2N}^{\mathrm{T}}\mathcal{L}_\triangle\hat{v}_{2N}. \tag{13b}$$

In view of (13a) and $\|\tilde{x}\|^2 = \|\hat{x}\|^2$, together with the s-strongly convexity of f_i in Assumption 2, one has

$$-k\tilde{x}^{\mathrm{T}}h \le -ks\|\hat{x}\|^2. \tag{14}$$

Similarly, due to $\|\tilde{y}\|^2 = \|\hat{y}\|^2$ and by the Cauchy Schwartz's inequality and the ℓ-Lipschitz continuous of the gradient of f_i in Assumption 3, it follows that

$$2k\tilde{y}^{\mathrm{T}}h \le k\|\hat{y}\|^2 + k\ell^2\|\hat{x}\|^2 \tag{15}$$

$$k\hat{z}_{2N}^{\mathrm{T}}(P^{\mathrm{T}}\otimes I_n)h \le \frac{k}{2}\|\hat{z}_{2N}\|^2 + \frac{k\ell^2}{2}\|\hat{x}\|^2. \tag{16}$$

Inserting (14)−(16) into (13a)−(13b), it follows that

$$\begin{aligned} \dot{V} \le& -\left(s - \frac{3\ell^2}{2}\right)k\hat{x}^{\mathrm{T}}\hat{x} + \hat{x}^{\mathrm{T}}(\hat{v}+\hat{y}) - \hat{v}^{\mathrm{T}}\hat{v} - 2\hat{v}^{\mathrm{T}}\hat{y} - \\ & (2-k)\hat{y}^{\mathrm{T}}\hat{y} + \hat{y}_{2N}^{\mathrm{T}}\mathcal{L}_\triangle\hat{y}_{2N} + \frac{k}{2}\hat{z}_{2N}^{\mathrm{T}}\hat{z}_{2N} + \\ & \hat{x}_{2N}^{\mathrm{T}}\mathcal{L}_\triangle\hat{z}_{2N} - \hat{z}_{2N}^{\mathrm{T}}\mathcal{L}_\triangle\hat{z}_{2N}. \end{aligned} \tag{17}$$

Considering $\lambda_2 I_{N-1} \le \mathcal{L}_\triangle \le \lambda_N I_{N-1}$, it can be checked that

$$-\begin{bmatrix} \hat{x}_{2N} \\ \hat{z}_{2N} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \frac{\lambda_N}{2}I_{N-1} & -\frac{\mathcal{L}_\triangle}{2}I_{N-1} \\ -\frac{\mathcal{L}_\triangle}{2}I_{N-1} & \frac{\mathcal{L}_\triangle}{2}I_{N-1} \end{bmatrix} \begin{bmatrix} \hat{x}_{2N} \\ \hat{z}_{2N} \end{bmatrix} \le 0$$

which can be equivalently simplified as

$$-\frac{\lambda_N}{2}\hat{x}_{2N}^{\mathrm{T}}\hat{x}_{2N} + \hat{x}_{2N}^{\mathrm{T}}\mathcal{L}_\triangle\hat{z}_{2N} - \frac{1}{2}\hat{z}_{2N}^{\mathrm{T}}\mathcal{L}_\triangle\hat{z}_{2N} \le 0. \tag{18}$$

By (18), adding and subtracting the term $\frac{\lambda_N}{2}\|\hat{x}_{2N}\|^2$ to the right side of (17), the time-derivative of V can be further estimated as

$$\begin{aligned} \dot{V} \le& -\left(s - \frac{3\ell^2}{2}\right)k\hat{x}^{\mathrm{T}}\hat{x} + \hat{x}^{\mathrm{T}}(\hat{v}+\hat{y}) - \hat{v}^{\mathrm{T}}\hat{v} - 2\hat{v}^{\mathrm{T}}\hat{y} - \\ & (2-k)\hat{y}^{\mathrm{T}}\hat{y} + \hat{y}_{2N}^{\mathrm{T}}\mathcal{L}_\triangle\hat{y}_{2N} + \frac{k}{2}\hat{z}_{2N}^{\mathrm{T}}\hat{z}_{2N} + \\ & \frac{\lambda_N}{2}\hat{x}_{2N}^{\mathrm{T}}\hat{x}_{2N} - \frac{1}{2}\hat{z}_{2N}^{\mathrm{T}}\mathcal{L}_\triangle\hat{z}_{2N}. \end{aligned} \tag{19}$$

Note that $-\hat{z}_{2N}^{\rm T}\mathcal{L}_{\triangle}\hat{z}_{2N} \le -\lambda_2\hat{z}_{2N}^{\rm T}\hat{z}_{2N}$ and $\hat{y}_{2N}^{\rm T}\mathcal{L}_{\triangle}\hat{y}_{2N} \le \lambda_N\hat{y}_{2N}^{\rm T}\hat{y}_{2N}$. In view of $\lambda_N\hat{y}_{2N}^{\rm T}\hat{y}_{2N} \le \lambda_N\hat{y}^{\rm T}\hat{y}$ and $\lambda_N\hat{x}_{2N}^{\rm T}\hat{x}_{2N} \le \lambda_N\hat{x}^{\rm T}\hat{x}$, one has

$$\dot{V} \le -\left[\left(s-\frac{3\ell^2}{2}\right)k-\frac{\lambda_N}{2}\right]\hat{x}^{\rm T}\hat{x}+\hat{x}^{\rm T}(\hat{v}+\hat{y})-\hat{v}^{\rm T}\hat{v}-2\hat{v}^{\rm T}\hat{y}-(2-\lambda_N-k)\hat{y}^{\rm T}\hat{y}-\frac{\lambda_2-k}{2}\hat{z}_{2N}^{\rm T}\hat{z}_{2N} \le$$
$$-\begin{bmatrix}\hat{x}\\ \hat{v}\\ \hat{y}\\ \hat{z}_{2N}\end{bmatrix}^{\rm T}\begin{bmatrix}\left[\left(s-\frac{3\ell^2}{2}\right)k-\frac{\lambda_N}{2}\right] & -\frac{1}{2} & -\frac{1}{2} & 0\\ -\frac{1}{2} & 1 & 1 & 0\\ -\frac{1}{2} & 1 & (2-\lambda_N-k) & 0\\ 0 & 0 & 0 & \frac{\lambda_2-k}{2}\end{bmatrix}\begin{bmatrix}\hat{x}\\ \hat{v}\\ \hat{y}\\ \hat{z}_{2N}\end{bmatrix}. \quad (20)$$

Denoted by M_1 the 4×4 matrix on the right side of (20), the remaining task is to prove that the matrix M_1 is positive definite. To this end, let

$$M_1 = \begin{pmatrix} A_1 & B_1\\ B_1^{\rm T} & C_1\end{pmatrix} \quad (21)$$

with

$$A_1 = \begin{bmatrix}[sk-(3\ell^2/2)k-\lambda_N/2] & -1/2\\ -1/2 & 1\end{bmatrix}$$
$$B_1 = \begin{bmatrix}-1/2 & 0\\ 1 & 0\end{bmatrix}$$
$$C_1 = \begin{bmatrix}(2-\lambda_N-k) & 0\\ 0 & (\lambda_2-k)/2\end{bmatrix}.$$

Some simple calculations show that $A_1>0$ and $C_1-B_1^{\rm T}A_1^{-1}B_1>0$ due to the condition (7). It follows from the Schur complement [46] that $M_1>0$. Noting that the equilibrium (x^*,v^*,y^*,z^*) of the dynamics (5) satisfies $x^*=1_N\otimes x^*$ with x^* as the optimal solution and referring to the orthogonal transformation in (10), it indicates that the equilibrium is asymptotically stable and consequently $\lim_{t\to\infty}x_i(t)=x^*$ for each $i\in\mathcal{V}$. □

Remark 4. Although the continuous-time distributed heavy-ball method is also studied in [41], the initial condition $\sum_{i=1}^N\tau_i(0)=0_m$ is required. The satisfaction of this initial condition depends on the information of all the agents, and thus it is global information. Furthermore, in some circumstances, the initial values satisfying $\sum_{i=1}^N\tau_i(0)=0_m$ do not obey the zero sum condition if agents are added or deleted from the network. Consequently, our proposed algorithm (3) removes the utilization of this initial condition. In addition, while [41] only deals with the relatively simple case of an undirected graph, our work also considers the case of an unbalanced directed graph, and we will illustrate later in Section 4.

Compared with first-order distributed gradient-based algorithms, the second-order distributed convex optimization algorithm (3) can guarantee a asymptotic convergence while having an faster convergence to the optimal solution x^*, which is illustrated later in Section 5.

4 Distributed convex optimization over directed graphs

Let us emphasize that, compared with the cases of undirected graphs, there are relatively few studies in the distributed convex optimization over directed graphs. Therefore, in this section, we extend the undirected graph analyzed in Section 3 to an unbalanced directed graph.

For each agent $i\in\mathcal{V}$, the distributed second-order optimization algorithm is defined as follows:

$$\begin{cases}\dot{x}_i = v_i & (22a)\\ \dot{v}_i = -kv_i-y_i+r_{ii}\sum\limits_{j\in N_i}(x_j-x_i) & (22b)\\ \dot{y}_i = -y_i+\nabla f_i(x_i)+r_{ii}\sum\limits_{j\in N_i}(z_j-z_i) & (22c)\\ \dot{z}_i = -r_{ii}\sum\limits_{j\in N_i}(y_j-y_i) & (22d)\\ \dot{r}_i = \sum\limits_{j\in N_i}(r_j-r_i) & (22e)\end{cases}$$

where $x_i,v_i,y_i,z_i\in\mathbf{R}^n$, $r_i\in\mathbf{R}^N$ and $r_{ii}\in\mathbf{R}$ is the i-th component of r_i. Also, k is a positive parameter to be designed. Considering that the left eigenvector of the Laplacian matrix L associated with the zero eigenvalues is unknown in advance, r_i is designed to be a local estimation of the left eigenvector. In order to achieve the estimation, we initialize $r_i(0)$ as $r_i(0)=(\underbrace{0,\cdots,0}_{i-1},1,\underbrace{0,\cdots,0}_{N-i})^{\rm T}$.

Remark 5. Compared with solving the convex optimization problem (1) under undirected graphs, the cases of directed graphs are more challenging due to the asymmetric Laplacian matrix L associated with directed graphs. As a consequence, the distributed algorithms under undirected graphs cannot be directly applied to directed graphs. In the existing literature, various first-order algorithms have been developed to solve the convex optimization problem (1) on balanced directed graphs[43, 47]. Besides, a few second-order algorithms are also designed for balanced directed graphs[35, 36]. However, when we consider the cases of unbalanced directed graphs, the

Laplacian matrix L of directed graphs no longer satisfies $1_N^{\rm T} L = 0_N$, which makes the algorithms proposed in balanced directed graphs unable to solve the cases of unbalanced directed graphs. Although distributed optimization algorithms are proposed for unbalanced directed graphs to solve the convex optimization problem (1) in [39, 40, 48], most of the algorithms are first-order type while the second-order algorithms that can achieve accelerated convergence are hardly studied. In an endeavor to design a second-order distributed convex optimization for an unbalanced directed graph, based on the algorithm (3), we use the equation (22e) and the additional information r_{ii} in (22b)−(22d) to estimate the left eigenvector of the Laplacian matrix L associated with the zero eigenvalues. Therefore, a distributed second-order algorithm under constant step size is proposed for unbalanced directed graphs and ensures the optimality conditions (2) at the optimal solution x^* of the convex optimization problem (1).

In order to analyze the convergence of the algorithm (22), we offer Lemmas 2–6 to help the analysis.

Lemma 2.[39] Suppose that the directed graph $\mathcal{G}$ satisfies Assumption 1, then the following statements hold.

1) There exists a positive left eigenvector $d = (d_1, \cdots, d_N)^{\rm T} \in \mathbf{R}^N$ of Laplacian matrix L associated with the zero eigenvalues such that $d_i > 0$ for $i \in \mathcal{V}$, $d^{\rm T} L = 0_N$ and $\sum_{i=1}^N d_i = 1$.

2) Denoting $D = {\rm diag}(d_1, \cdots, d_N)$, then $\hat{L} \triangleq \dfrac{1}{2}(DL + L^{\rm T} D)$ is a symmetric positive semi-definite matrix. For convenience, we denote $\mathcal{D} \triangleq D \otimes I_n$ and $\hat{\mathcal{L}} \triangleq \hat{L} \otimes I_n$.

Lemma 3.[49] Suppose that the directed graph $\mathcal{G}$ satisfies Assumption 1 and the Laplacian matrix L satisfies $Lw_r = 0$, $w_l^{\rm T} L = 0$, and $w_l^{\rm T} w_r = 1$, where w_r and w_l denote the right and left eigenvectors of Laplacian matrix L, respectively. Then $\lim_{t\to\infty} \exp(-Lt) = w_r^{\rm T} w_l$, where $\exp(-Lt), t > 0$ is a non-negative matrix with positive diagonal entries.

To facilitate the convergence analysis, the algorithm (22) can be rewritten into the following compact form:

$$\begin{cases} \dot{x} = v \\ \dot{v} = -kv - y - \mathcal{R}\mathcal{L}x \\ \dot{y} = -y + G(x) - \mathcal{R}\mathcal{L}z \\ \dot{z} = \mathcal{R}\mathcal{L}y \\ \dot{r} = -\mathbb{L}r \end{cases} \tag{23}$$

where $\mathbb{L} \triangleq L \otimes I_N$, $x, v, y, z \in \mathbf{R}^{nN}$, $r = {\rm col}(r_1, \cdots, r_N) \in \mathbf{R}^{NN}$ and $\mathcal{R} \triangleq R \otimes I_n$ with $R = {\rm diag}(r_{11}, \cdots, r_{NN}) \in \mathbf{R}^{N\times N}$. In this paper, the role of r in the algorithm (23) is to estimate the left eigenvector d of the Laplacian matrix L associated with the zero eigenvalues so that the equilibrium and the convergence of the algorithm (23) can be analyzed. Now we first show that r in (23) converges to the left eigenvector d in Lemma 4.

Lemma 4. For the convex optimization problem (1) under Assumptions 1−3 and Definition 1, the trajectory of r_i in (22) with any initial condition $r_i(0)$ satisfying $r_i(0) = (\underbrace{0, \cdots, 0}_{i-1}, 1, \underbrace{0, \cdots, 0}_{N-i})^{\rm T}$ achieves $\lim_{t\to\infty} r_i = d$ for each $i \in \mathcal{V}$.

Proof. From the last equation $\dot{r} = -\mathbb{L}r$ in the dynamics (23), one has that $r = \exp(-\mathbb{L}t)r(0)$. In view of Lemma 2, we obtain d, which is a positive left eigenvector of the Laplacian matrix L associated with the zero eigenvalues and satisfies $d^{\rm T} L = 0_N$ and $d^{\rm T} 1_N = 1$. Since 1_N is the right eigenvector of the Laplacian matrix L, we can obtain from Lemma 3 that

$$\begin{aligned} \lim_{t\to\infty} r =& \lim_{t\to\infty} \exp(-bfLt) r(0) = \\ & (1_N d^{\rm T} \otimes I_N) r(0) = 1_N \otimes d. \end{aligned} \tag{24}$$

The equation (24) indicates that $\lim_{t\to\infty} r_i = d$ and therefore $\lim_{t\to\infty} R = D$, with D defined in Lemma 2. □

Based on Lemma 4, we then analyze the equilibrium $(x^*, v^*, y^*, z^*, r^*)$ of the dynamics (23).

Lemma 5. For the convex optimization problem (1) under Assumptions 1−3 and Definition 1, let $(x^*, v^*, y^*, z^*, r^*)$ be an equilibrium of the dynamics (23). Then $(x^*, v^*, y^*, z^*, r^*) = (1_N \otimes x^*, 0_{nN}, 0_{nN}, z^*, 1_N \otimes d)$ with z^* arbitrarily satisfies $\mathcal{D}\mathcal{L}z^* = G(x^*)$.

Proof. In view of Lemma 4, we obtain $r^* = 1_N \otimes d$ and $R^* = D$, with d and D defined in Lemma 2. Since $(x^*, v^*, y^*, z^*, r^*)$ is the equilibrium of the dynamics (23), we substitute it into the dynamics (23) to obtain

$$\begin{cases} 0_{nN} = v^* & \text{(25a)} \\ 0_{nN} = -kv^* - y^* - \mathcal{D}\mathcal{L}x^* & \text{(25b)} \\ 0_{nN} = -y^* + G(x^*) - \mathcal{D}\mathcal{L}z^* & \text{(25c)} \\ 0_{nN} = \mathcal{D}\mathcal{L}y^* & \text{(25d)} \\ 0_{NN} = -\mathbb{L}(1_N \otimes d). & \text{(25e)} \end{cases}$$

Similar to the analysis in Lemma 1, we can conclude that $(x^*, v^*, y^*, z^*, r^*) = (1_N \otimes x^*, 0_{nN}, 0_{nN}, z^*, 1_N \otimes d)$ with $\mathcal{D}\mathcal{L}z^* = G(x^*)$. □

In order to analyze the convergence of the dynamics (23) to the equilibrium, we first give Lemma 6 to help the analysis.

Lemma 6. Let Assumptions 1−3 and Definition 1 hold and assume that the Lipschitz parameter ℓ and the minimal non-zero eigenvalue $\lambda_2(\hat{L})$ of $\hat{L}$ satisfy $\lambda_2(\hat{L}) > 1 + \ell$ so that there exists a parameter k such that

$$2\lambda_2(\hat{L}) - \ell - q < k < 2\lambda_2(\hat{L}) - \ell + q \tag{26}$$

where $q = \dfrac{\sqrt{\lambda_2(\hat{L}) - 1 - \ell}}{2}$. Then, the mapping $F(x, v, y)$ defined as

$$F(x,v,y)=\begin{bmatrix}k\hat{\mathcal{L}}x+ky\\ kv+2y\\ y-G(x)\end{bmatrix} \tag{27}$$

is μ-strongly monotone for some $\mu\in\mathbf{R}_{>0}$.

Proof. The mapping $F(x,v,y)$ is μ-strongly monotone if

$$\begin{aligned}&\mathrm{col}(x-x',v-v',y-y')^{\mathrm{T}}[F(x,v,y)-F(x',v',y')]\geq\\&\quad\mu\|x-x'\|^2+\mu\|v-v'\|^2+\mu\|y-y'\|^2.\end{aligned} \tag{28}$$

In order to prove (28), we substitute (27) into the left side of (28), one has that

$$\begin{aligned}&\mathrm{col}(x-x',v-v',y-y')^{\mathrm{T}}[F(x,v,y)-F(x',v',y')]=\\&\quad k(x-x')^{\mathrm{T}}\hat{\mathcal{L}}(x-x')+k(x-x')^{\mathrm{T}}(y-y')+\\&\quad k(v-v')^{\mathrm{T}}(v-v')+2(v-v')^{\mathrm{T}}(y-y')+\\&\quad (y-y')^{\mathrm{T}}(y-y')-(y-y')^{\mathrm{T}}[G(x)-G(x')].\end{aligned} \tag{29}$$

Note that $(v-v')^{\mathrm{T}}(y-y')\geq-\|v-v'\|\|y-y'\|$ and $(x-x')^{\mathrm{T}}(y-y')\geq-\|x-x'\|\|y-y'\|$. By Lemma 2, we obtain $(x-x')^{\mathrm{T}}\hat{\mathcal{L}}(x-x')\geq\lambda_2(\hat{L})\|x-x'\|^2$ when we assume that $x-x'\neq 1_N\otimes a$ for some $a\in\mathbf{R}^n$. Besides, by the ℓ-Lipschitz continuous of the gradient of f_i in Assumption 3, it follows that

$$\begin{aligned}&-(y-y')^{\mathrm{T}}[G(x)-G(x')]\geq-\|y-y'\|\|G(x)-G(x')\|\geq\\&-\ell\|y-y'\|\|x-x'\|.\end{aligned}$$

Then, (29) can be calculated as

$$\begin{aligned}&\mathrm{col}(x-x',v-v',y-y')^{\mathrm{T}}[F(x,v,y)-F(x',v',y')]\geq\\&\quad\lambda_2(\hat{L})k\|x-x'\|^2-(k+\ell)\|y-y'\|\|x-x'\|+\\&\quad k\|v-v'\|^2+\|y-y'\|^2-2\|v-v'\|\|y-y'\|=\\&\quad\begin{bmatrix}\|x-x'\|\\ \|v-v'\|\\ \|y-y'\|\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\lambda_2(\hat{L})k & 0 & -\dfrac{k+\ell}{2}\\ 0 & k & -1\\ -\dfrac{k+\ell}{2} & -1 & 1\end{bmatrix}\begin{bmatrix}\|x-x'\|\\ \|v-v'\|\\ \|y-y'\|\end{bmatrix}.\end{aligned} \tag{30}$$

It is obvious that (28) holds if and only if the 3×3 matrix in (30) is positive definiteness. Some simple calculations show that the 3×3 matrix is positive definiteness due to the condition (26). Thus, (28) holds, i.e., the mapping $F(x,v,y)$ is μ-strongly monotone. □

Theorem 2. For the convex optimization problem (1) under Assumptions 1−3 and Definition 1, assume that ℓ and $\lambda_2(\hat{L})$ satisfy $\lambda_2(\hat{L})>1+\ell$ so that there exists a parameter k that satisfies condition (26). Then, the trajectory of dynamics (22) with any initial condition $(x_i(0),\ v_i(0),\ y_i(0),\ z_i(0),\ r_i(0))$ satisfying $r_i(0)=(\underbrace{0,\cdots,0}_{i-1},1,\underbrace{0,\cdots,0}_{N-i})^{\mathrm{T}}$ achieve $x_i(t)\to x^*$ asymptotically as $t\to\infty$ for each $i\in\mathcal{V}$.

Proof. To facilitate the subsequent analysis, we set $w=\mathrm{col}(x,v,y,z)$, and the dynamics of x,v,y,z in (23) can be rewritten in terms of w as

$$\underbrace{\begin{pmatrix}\dot{x}\\ \dot{v}\\ \dot{y}\\ \dot{z}\end{pmatrix}}_{\dot{w}}=\underbrace{\begin{pmatrix}v\\ -kv-y-\mathcal{DL}x\\ -y+G(x)-\mathcal{DL}z\\ \mathcal{DL}y\end{pmatrix}}_{f(w)}+\underbrace{\begin{pmatrix}0_{nN}\\ (\mathcal{D}-\mathcal{R})\mathcal{L}x\\ (\mathcal{D}-\mathcal{R})\mathcal{L}z\\ (\mathcal{R}-\mathcal{D})\mathcal{L}y\end{pmatrix}}_{g(w,t)}. \tag{31}$$

Thus, we need to analyze the stability of system (31). Due to the analysis of Lemma 4, one has $\lim_{t\to\infty}g(w,t)=0$. Now, we first prove the stability of $\dot{w}=f(w)$, namely

$$\begin{cases}\dot{x}=v\\ \dot{v}=-kv-y-\mathcal{DL}x\\ \dot{y}=-y+G(x)-\mathcal{DL}z\\ \dot{z}=\mathcal{DL}y.\end{cases} \tag{32}$$

Note that the equilibrium of (32) is the same as the dynamics (23). Then, we transfer the equilibrium (x^*, v^*,y^*,z^*) of the dynamics (23) into the origin by letting $\widetilde{x}=x-x^*$, $\widetilde{v}=v-v^*$, $\widetilde{y}=y-y^*$, $\widetilde{z}=z-z^*$. The dynamics (32) under the new coordinates can be described as

$$\begin{cases}\dot{\widetilde{x}}=\widetilde{v}\\ \dot{\widetilde{v}}=-k\widetilde{v}-\widetilde{y}-\mathcal{DL}\widetilde{x}\\ \dot{\widetilde{y}}=-\widetilde{y}+h-\mathcal{DL}\widetilde{z}\\ \dot{\widetilde{z}}=\mathcal{DL}\widetilde{y}.\end{cases} \tag{33}$$

Thus, we only need to analyze the stability of the zero solution of (33). We construct a Lyapunov candidate V, with V given below:

$$V=\frac{1}{2}\|k\widetilde{x}+\widetilde{v}\|^2+\frac{1}{2}\|\widetilde{v}\|^2+\frac{1}{2}\|\widetilde{y}\|^2+\frac{1}{2}\|\widetilde{z}\|^2+\widetilde{x}^{\mathrm{T}}\hat{\mathcal{L}}\widetilde{x}. \tag{34}$$

In view of $\hat{L}\triangleq\frac{1}{2}(DL+L^{\mathrm{T}}D)$ in Lemma 2, the time-derivative of V along the trajectories of (33) can be calculated as

$$\begin{aligned}\dot{V}=&-k\widetilde{x}^{\mathrm{T}}\hat{\mathcal{L}}\widetilde{x}-k\widetilde{x}^{\mathrm{T}}\widetilde{y}-k\widetilde{v}^{\mathrm{T}}\widetilde{v}-2\widetilde{v}^{\mathrm{T}}\widetilde{y}-\widetilde{y}^{\mathrm{T}}\widetilde{y}+\widetilde{y}^{\mathrm{T}}h=\\&-(\widetilde{x}^{\mathrm{T}},\widetilde{v}^{\mathrm{T}},\widetilde{y}^{\mathrm{T}})\begin{bmatrix}k\hat{\mathcal{L}}\widetilde{x}+k\widetilde{y}\\ k\widetilde{v}+2\widetilde{y}\\ \widetilde{y}-h\end{bmatrix}.\end{aligned} \tag{35}$$

In view of (27) of $F(x,v,y)$, the time-derivative of V can be further calculated as

$$\dot{V} = -(\tilde{x}^{\mathrm{T}}, \tilde{v}^{\mathrm{T}}, \tilde{y}^{\mathrm{T}})[F(x, v, y) - F(x^*, v^*, y^*)]. \quad (36)$$

By the μ-strongly monotone of $F(x, v, y)$ and (28) proved in Lemma 6, the time-derivative of V can be estimated as

$$\dot{V} \le -\mu(\|\tilde{x}\|^2 + \|\tilde{v}\|^2 + \|\tilde{y}\|^2) \le 0. \quad (37)$$

Applying Lasalle's invariance principle, we conclude that $\tilde{x}, \tilde{v}, \tilde{y}$ all converge to zero solution. Thus, we obtain $x = x^*$, $v = v^*$, $y = y^*$. Inserting these results into the right-hand side of the dynamic (32), one has $\mathcal{D}\mathcal{L}z = G(x^*)$. Therefore, the equilibrium of (32) is asymptotically stable.

Based on the stability of (32), we then study the stability of the system (31). In fact, system (31) can be regarded as the perturbed system of (32). Recall the fact that if the equilibrium of (32) is asymptotically stable and the perturbation term $g(w, t)$ satisfies $\lim_{t\to\infty} g(w, t) = 0$, then the equilibrium of (31) is asymptotically stable[50]. Thus, the equilibrium of the dynamics (23) is asymptotically stable and consequently $\lim_{t\to\infty} x_i(t) = x^*$ for each $i \in \mathcal{V}$. □

Remark 6. In the proof of Theorem 2, we assume that $x - x^* \ne 1_N \otimes a$ for $a \in \mathbf{R}^n$. Then, we further show that the result of Theorem 2 still holds when we assume that $x - x^* = 1_N \otimes a$. In view of Lemma 4, we show that r in (23) converges to the left eigenvector d of the Laplacian matrix L associated with the zero eigenvalues as $t \to \infty$. Inserting this result and $x - x^* = 1_N \otimes a$ into the dynamics (23), we consequently give $x = x^*, v = y = 0_{nN}$ and $\mathcal{D}\mathcal{L}z = G(x^*)$. Therefore, we just need to consider $x - x^* \ne 1_N \otimes a$ for the proof of Theorem 2.

Remark 7. The fundamental algorithm in distributed optimization is the famous approach of distributed gradient descent with diminishing step sizes, where only the state variable is transmitted in the network. Despite its simplicity, this algorithm has its own disadvantage, such as the slow convergence due to diminishing step sizes. In our work, to improve the convergence speed, we depart from adopting the above first-order algorithmic structure. Instead, we use a second-order set up with the hope to achieve potential faster convergence, merge the gradient tracking strategy into our algorithms (3) and (22) to ensure the satisfaction of the optimality condition (2), and abandon the use of diminishing step sizes to avoid slow convergence. However, these improvements come at the cost of requiring more intermediate variables to be transmitted in the network.

5 Simulation

In this section, we illustrate the convergence performances of algorithm (3) and algorithm (22) with two specific examples. We first consider a networked optimization problem (1), with each local cost function given by

$$f_1(x) = \frac{1}{2}\mathrm{e}^{-\frac{2}{5}x} + \frac{1}{10}x, \quad f_2(x) = x^2 + \mathrm{e}^{\frac{1}{10}x},$$
$$f_3(x) = \frac{1}{5}x^2 + \frac{1}{2}x, \quad f_4(x) = \frac{1}{5}\mathrm{e}^{-\frac{1}{5}x} + \mathrm{e}^{\frac{2}{5}x},$$
$$f_5(x) = \frac{3}{20}x^2 - \frac{3}{10}x.$$

These local cost functions are twice-differentiable and strongly convex, and the gradients of f_i are Lipschitz continuous. Thus, f_i satisfies Assumptions 2 and 3 for $i = 1, 2, 3, 4, 5$. Besides, the interaction of the network is represented by an undirected graph $\mathcal{G}$ in Fig. 1, whose Laplacian matrix L is given by

$$L = \frac{1}{8}\begin{bmatrix} 3 & -1 & -1 & -1 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 2 & -1 & 0 \\ -1 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

with eigenvalues $\lambda_1 = 0$, $\lambda_2 = \frac{109}{1\,250}$, $\lambda_3 = \frac{1\,727}{10\,000}$, $\lambda_4 = \frac{4\,523}{10\,000}$ and $\lambda_5 = \frac{2\,689}{5\,000}$. We set the initial conditions as $x(0) = [3, -1, -2, 0, 2]^{\mathrm{T}}$, $v(0) = [1, -1, 0, 0, 0]^{\mathrm{T}}$, $y(0) = [-2, 0, 2, 1, -1]^{\mathrm{T}}$ and $z(0) = [0, 0, 0, 0, 0]^{\mathrm{T}}$, and choose $k = \frac{1}{8}$. It can be checked that the condition (7) in Theorem 1 is satisfied.

The simulation result of the algorithm (3) is shown in Fig. 2. Fig. 2 shows that each $x_i(t)$ converges to the solution point $x^* = -\frac{189}{1\,000}$ of the global cost function $f(x) = \frac{1}{5}\sum_{i=1}^{5} f_i(x)$ in the algorithm (3). Also, each $y_i(t)$ that tracks the average gradients $\frac{1}{5}\sum_{i=1}^{5} \nabla f_i(x_i(t))$ converges to 0.

In addition, we illustrate the fast convergence performance of the algorithm (3) compared with the first-order distributed optimization algorithm. Firstly, the first-order distributed optimization algorithm in [51] is shown as follows:

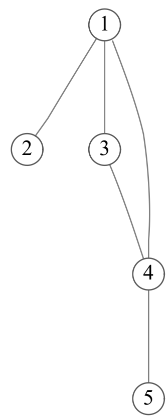


Fig. 1 An undirected graph of five agents

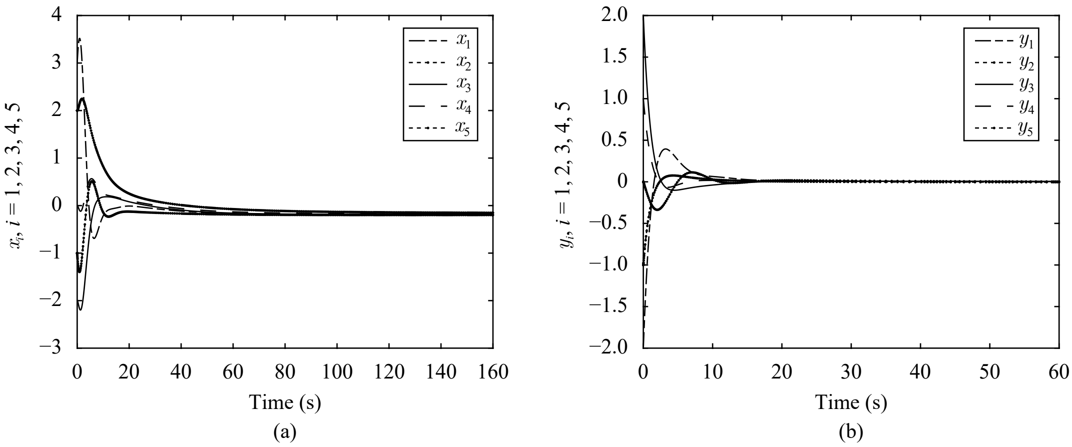


Fig. 2 Time evolution of the state x and state y in the algorithm (3): (a) Each component of x in the algorithm (3) converges to $-\dfrac{189}{1\,000}$ in 160 s; (b) Each component of y in the algorithm (3) converges to 0.

$$\begin{aligned}\dot{x}_i &= \sum\nolimits_{j\in N_i}(x_j - x_i) + \sum\nolimits_{j\in N_i}(z_j - z_i) - \nabla f_i(x_i)\\ \dot{z}_i &= -\sum\nolimits_{j\in N_i}(x_j - x_i).\end{aligned} \tag{38}$$

We apply the algorithm (38) to solve the convex optimization problem (1) with the same conditions and initial values. The result is shown in Fig. 3. From the results of Figs. 2 and 3, it is easy to see that the converging of our algorithm in (3) is 160 s, which is shorter than 300 s in (38).

Then we consider the convergence performance of the algorithm (22). The interaction of the network with four agents are depicted by the strongly connected unbalanced directed graph in Fig. 4, and the cost function f_i, $i = 1, 2, 3, 4$ in the convex optimization problem (1) are defined as follows:

$$\begin{aligned} f_1(x) &= 2\mathrm{e}^{-\frac{1}{5}x} + \frac{3}{10}x, \quad f_2(x) = x^2 + 2\mathrm{e}^{\frac{1}{2}x},\\ f_3(x) &= \frac{1}{4}x^2 - \frac{1}{10}x, \quad f_4(x) = \frac{1}{5}x^2 + \frac{1}{2}x.\end{aligned}$$

It can be checked that the cost function f_i satisfies Assumptions 2 and 3 for $i = 1, 2, 3, 4$. Besides, the Laplacian matrix L of the unbalanced directed graph is given by

$$L = \begin{bmatrix} 2 & -1 & -1 & 0\\ -1 & 2 & -1 & 0\\ -1 & 0 & 2 & -1\\ -1 & 0 & 0 & 1\end{bmatrix}.$$

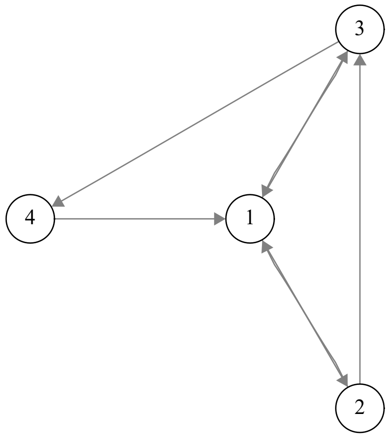


Fig. 4 An unbalanced directed graph of four agents

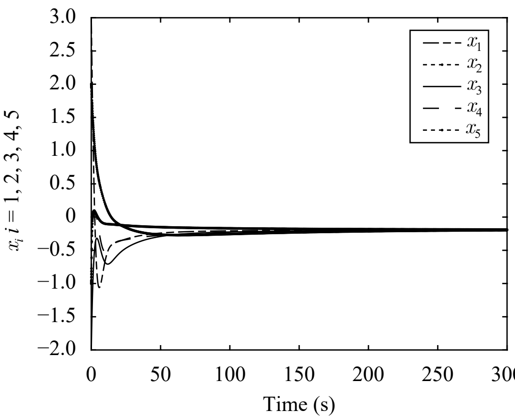


Fig. 3 Each $x_i(t)$ in the algorithm (38) converges to $-\dfrac{189}{1\,000}$ in 300 s

With the initial condition $r_i(0)$ defined in the algorithm (22), we set other initial conditions as $x(0) = [3, -1, -2, 0]^{\mathrm{T}}$, $v(0) = [1, -1, 0, 0]^{\mathrm{T}}$, $y(0) = [-2, 0, 2, 1]^{\mathrm{T}}$and $z(0) = [0, 0, 0, 0]^{\mathrm{T}}$, and choose $k = 4$. It can be checked that the condition (26) in Lemma 6 is satisfied.

The simulation results of the algorithm (22) are shown in Figs. 5 and 6. Fig. 5 shows that each $x_i(t)$ converges to the solution point $x^* = -\dfrac{189}{500}$ of the global cost function $f(x) = \dfrac{1}{4}\sum_{i=1}^{4} f_i(x)$ in the algorithm (22). Also, each $y_i(t)$ that tracks the average gradients $\dfrac{1}{4}\sum_{i=1}^{4}\nabla f_i(x_i(t))$ converges to 0. In Fig. 6, each $r_i(t)$ converges to the left eigenvector $\left(\dfrac{1}{3}, \dfrac{1}{6}, \dfrac{1}{4}, \dfrac{1}{4}\right)^{\mathrm{T}}$ of the Laplacian matrix L associated with the zero eigenvalue.

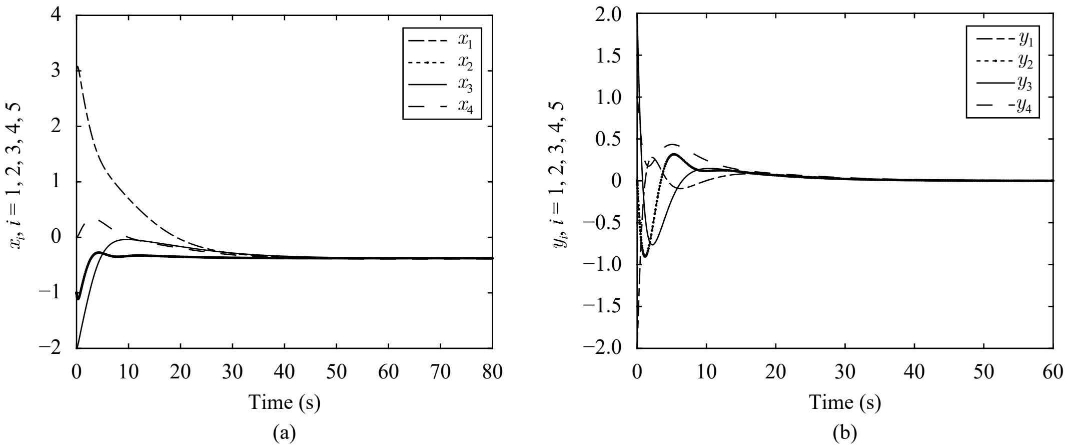


Fig. 5 Time evolution of the state x and state y in the algorithm (22): (a) Each component of x in the algorithm (22) converges to $-\frac{189}{500}$ in 60 s; (b) Each component of y in the algorithm (22) converges to 0.

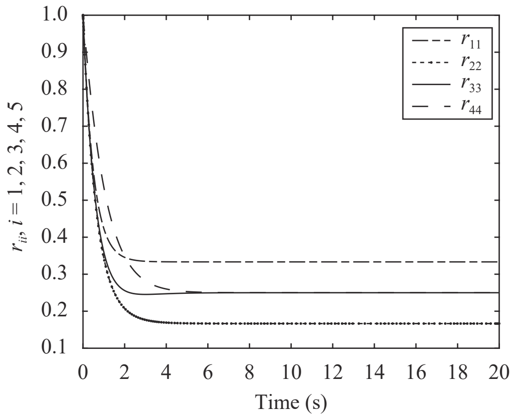


Fig. 6 Each component of r in the algorithm (22) converges to $\left(\frac{1}{3}, \frac{1}{6}, \frac{1}{4}, \frac{1}{4}\right)^{\rm T}$.

6 Conclusion and future work

In this paper, we design continuous-time distributed second-order optimization algorithms to solve the convex optimization problem over undirected and unbalanced directed graphs. Incorporating the estimation scheme of the left eigenvector of the Laplacian matrix associated with the zero eigenvalues, we successfully modified our initial algorithm to apply it to the distributed convex optimization over unbalanced directed graphs. Moreover, by relating the optimal solution to the equilibrium of our proposed optimization algorithms and referring to the strategy of gradient tracking, our algorithms can ensure that all agents asymptotically converge to the optimal solution of the algorithms under constant step size. Future, work may be directed towards applying our second-order algorithms to distributed convex optimization problems with equality and inequality constraints. Meanwhile, the main challenge of this future work is to analyze the convergence and run the simulation in the presence of gradient projection.

Acknowledgements

This work was supported by National Nature Science Foundation of China (Nos. 61663026, 62066026, 61963028 and 61866023) and Jiangxi NSF (No. 20192BAB 207025).

References

[1] W. Wang, J. S. Huang, C. Y. Wen, H. J. Fan. Distributed adaptive control for consensus tracking with application to formation control of nonholonomic mobile robots. *Automatica*, vol. 50, no. 4, pp. 1254–1263, 2014. DOI: 10.1016/j.automatica.2014.02.028.

[2] T. J. Zhang. Unmanned aerial vehicle formation inspired by bird flocking and foraging behavior. *International Journal of Automation and Computing*, vol. 15, no. 4, pp. 402–416, 2018. DOI: 10.1007/s11633-017-1111-x.

[3] Z. H. Deng, S. Liang, Y. G. Hong. Distributed continuous-time algorithms for resource allocation problems over weight-balanced digraphs. *IEEE Transactions on Cybernetics*, vol. 48, no. 11, pp. 3116–3125, 2018. DOI: 10.1109/TCYB.2017.2759141.

[4] M. Panda, B. Das, B. Subudhi, B. B. Pati. A comprehensive review of path planning algorithms for autonomous underwater vehicles. *International Journal of Automation and Computing*, vol. 17, no. 3, pp. 321–352, 2020. DOI: 10.1007/s11633-019-1204-9.

[5] Z. Wang, D. Wang, D. B. Gu. Distributed optimal state consensus for multiple circuit systems with disturbance rejection. *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2926–2939, 2020. DOI: 10.1109/TNSE.2020.3007472.

[6] W. J. Feng, R. Jiang, G. L. Liu. Distributed power control in cooperative cognitive ad hoc networks. *International Journal of Automation and Computing*, vol. 11, no. 4, pp. 412–417, 2014. DOI: 10.1007/s11633-014-0807-4.

[7] W. Chen, Y. F. Fu. Cooperative distributed target tracking algorithm in mobile wireless sensor networks. *Journal of Control Theory and Applications*, vol. 9, no. 2, pp. 155, vol. 9, no. 2, Article number 155, 2011. DOI: 10.1007/s11768-011-8124-8.

[8] T. Yang, X. L. Yi, J. F. Wu, Y. Yuan, D. Wu, Z. Y. Meng,

Y. G. Hong, H. Wang, Z. L. Lin, K. H. Johansson. A survey of distributed optimization. *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019. DOI: 10.1016/j.arcontrol.2019.05.006.

[9] I. Lobel, A. Ozdaglar. Distributed subgradient methods for convex optimization over random networks. *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1291–1306, 2011. DOI: 10.1109/TAC.2010.2091295.

[10] A. Nedic, A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009. DOI: 10.1109/TAC.2008.2009515.

[11] J. C. Duchi, A. Agarwal, 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. DOI: 10.1109/TAC.2011.2161027.

[12] A. Nedic, A. Ozdaglar, P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010. DOI: 10.1109/TAC.2010.2041686.

[13] K. I. Tsianos, S. Lawlor, M. G. Rabbat. Push-sum distributed dual averaging for convex optimization. In *Proceedings of the 51st IEEE Conference on Decision and Control*, IEEE, Maui, USA, pp. 5453–5458, 2012. DOI: 10.1109/CDC.2012.6426375.

[14] C. G. Xi, U. A. Khan. Distributed subgradient projection algorithm over directed graphs. *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3986–3992, 2017. DOI: 10.1109/TAC.2016.2615066.

[15] D. Jakovetić, J. Xavier, J. M. F. Moura. Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1131–1146, 2014. DOI: 10.1109/TAC.2014.2298712.

[16] S. Lee, A. Nedić. Asynchronous gossip-based random projection algorithms over networks. *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 953–968, 2016. DOI: 10.1109/TAC.2015.2460051.

[17] I. Lobel, A. Ozdaglar. Convergence analysis of distributed subgradient methods over random networks. In *Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing*, IEEE, Monticello, USA, pp. 353–360, 2008. DOI: 10.1109/ALLERTON.2008.4797579.

[18] K. Yuan, Q. Ling, W. T. Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016. DOI: 10.1137/130943170.

[19] I. Matei, J. S. Baras. Performance evaluation of the consensus-based distributed subgradient method under random communication topologies. *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 754–771, 2011. DOI: 10.1109/JSTSP.2011.2120593.

[20] J. M. Xu, S. Y. Zhu, Y. C. Soh, L. H. Xie. Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. In *Proceedings of the 54th IEEE Conference on Decision and Control*, IEEE, Osaka, Japan, pp. 2055–2060, 2015. DOI: 10.1109/CDC.2015.7402509.

[21] S. Pu, W. Shi, J. M. Xu, A. Nedić. Push-pull gradient methods for distributed optimization in networks. *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 1–16, 2021. DOI: 10.1109/TAC.2020.2972824.

[22] A. Nedić, A. Olshevsky, W. Shi, C. A. Uribe. Geometrically convergent distributed optimization with uncoordinated step-sizes. In *Proceedings of American Control Conference*, IEEE, Seattle, USA, pp. 3950–3955, 2017. DOI: 10.23919/ACC.2017.7963560.

[23] G. P. Chen, P. Yi, Y. G. Hong. Distributed optimization with projection-free dynamics. [Online], Available: https://arxiv.org/abs/2105.02450, 2021.

[24] M. J. Ye, G. Q. Hu, L. H. Xie, S. Y. Xu. Differentially private distributed Nash equilibrium seeking for aggregative games. *IEEE Transactions on Automatic Control*, to be published. DOI: 10.1109/TAC.2021.3075183.

[25] R. A. Freeman, P. Yang, K. M. Lynch. Stability and convergence properties of dynamic average consensus estimators. In *Proceedings of the 45th IEEE Conference on Decision and Control*, IEEE, San Diego, USA, pp. 338–343, 2006. DOI: 10.1109/CDC.2006.377078.

[26] M. J. Ye, G. Q. Hu. Game design and analysis for price-based demand response: An aggregate game approach. *IEEE Transactions on Cybernetics*, vol. 47, no. 3, pp. 720–730, 2017. DOI: 10.1109/TCYB.2016.2524452.

[27] J. Koshal, A. Nedić, U. V. Shanbhag. Distributed algorithms for aggregative games on graphs. *Operations Research*, vol. 64, no. 3, pp. 680–704, 2016. DOI: 10.1287/opre.2016.1501.

[28] X. He, D. W. C. Ho, T. W. Huang, J. Z. Yu, H. Abu-Rub, C. J. Li. Second-order continuous-time algorithms for economic power dispatch in smart grids. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 9, pp. 1482–1492, 2018. DOI: 10.1109/TSMC.2017.2672205.

[29] A. J. Wang, T. Dong, X. F. Liao. Distributed optimal consensus algorithms in multi-agent systems. *Neurocomputing*, vol. 339, pp. 26–35, 2019. DOI: 10.1016/j.neucom.2019.01.044.

[30] Y. S. Li, H. G. Zhang, B. N. Huang, J. Han. A distributed Newton-Raphson-based coordination algorithm for multi-agent optimization with discrete-time communication. *Neural Computing and Applications*, vol. 32, no. 9, pp. 4649–4663, 2020. DOI: 10.1007/s00521-018-3798-1.

[31] N. T. Tran, Y. W. Wang, W. Yang. Distributed optimization problem for double-integrator systems with the presence of the exogenous disturbance. *Neurocomputing*, vol. 272, pp. 386–395, 2018. DOI: 10.1016/j.neucom.2017.07.005.

[32] Z. H. Deng, S. Liang, W. Y. Yu. Distributed optimal resource allocation of second-order multiagent systems. *International Journal of Robust and Nonlinear Control*, vol. 28, no. 14, pp. 4246–4260, 2018. DOI: 10.1002/rnc.4233.

[33] D. Varagnolo, F. Zanella, A. Cenedese, G. Pillonetto, L. Schenato. Newton-raphson consensus for distributed convex optimization. *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 994–1009, 2016. DOI: 10.1109/TAC.2015.2449811.

[34] H. Attouch, X. Goudou, P. Redont. The heavy ball with friction method, I. the continuous dynamical system: Global exploration of the local minima of a real-valued function by asymptotic analysis of a dissipative dynamical system. *Communications in Contemporary Mathematics*, vol. 2, no. 1, pp. 1–34, 2000. DOI: 10.1142/S0219199700000025.

[35] S. L. Li, X. H. Nian, Z. H. Deng. Distributed optimization of second-order multi-agent systems with external disturbance over weight-balanced digraphs. In *Proceedings of*

Chinese Control Conference, IEEE, Guangzhou, China, pp. 2006–2011, 2019. DOI: 10.23919/ChiCC.2019.8865509.

[36] Z. H. Deng. Distributed algorithm design for resource allocation problems of second-order multiagent systems over weight-balanced digraphs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3512–3521, 2021. DOI: 10.1109/TSMC.2019.2930672.

[37] D. Wang, J. J. Yin, W. Wang. Distributed randomized gradient-free optimization protocol of multiagent systems over weight-unbalanced digraphs. *IEEE Transactions on Cybernetics*, vol. 51, no. 1, pp. 473–482, 2021. DOI: 10.1109/TCYB.2018.2890140.

[38] D. Wang, J. L. Yin, W. Wang. Design of fixed step-size distributed optimization protocol of multiagent systems over weighted unbalanced digraphs. In *Proceedings of the 8th International Conference on Information Science and Technology*, IEEE, Cordoba, Spain, pp. 321–328, 2018. DOI: 10.1109/ICIST.2018.8426143.

[39] Z. H. Li, Z. T. Ding, J. Y. Sun, Z. K. Li. Distributed adaptive convex optimization on directed graphs via continuous-time algorithms. *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1434–1441, 2018. DOI: 10.1109/TAC.2017.2750103.

[40] Y. N. Zhu, W. W. Yu, G. H. Wen, W. Ren. Continuous-time coordination algorithm for distributed convex optimization over weight-unbalanced directed networks. *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 7, pp. 1202–1206, 2019. DOI: 10.1109/TCSII.2018.2878250.

[41] W. Y. Yu, P. Yi, Y. G. Hong. A gradient-based dissipative continuous-time algorithm for distributed optimization. In *Proceedings of the 35th Chinese Control Conference*, IEEE, Chengdu, China, pp. 7908–7912, 2016. DOI: 10.1109/ChiCC.2016.7554612.

[42] C. Godsil, G. Royle. *Algebraic Graph Theory*. New York, USA: Springer, 2001. DOI: 10.1007/978-1-4613-0163-9.

[43] S. S. Kia, J. Cortés, S. Martínez. Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication. *Automatica*, vol. 55, pp. 254–264, 2015. DOI: 10.1016/j.automatica.2015.03.001.

[44] R. T. Rockafellar. *Convex Analysis*. Princeton, USA: Princeton University Press, 1970.

[45] R. Xin, 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. DOI: 10.1109/TAC.2019.2942513.

[46] F. Z. Zhang. *The Schur Complement and its Applications* Boston, USA: Springer, 2005. DOI: 10.1007/b105056.

[47] B. Gharesifard, J. Cortés. Distributed continuous-time convex optimization on weight-balanced digraphs. *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 781–786, 2014. DOI: 10.1109/TAC.2013.2278132.

[48] Y. N. Zhu, W. Ren, W. W. Yu, G. H. Wen. Distributed resource allocation over directed graphs via continuous-time algorithms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 2, pp. 1097–1106, 2021. DOI: 10.1109/TSMC.2019.2894862.

[49] R. Olfati-Saber, R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004. DOI: 10.1109/TAC.2004.834113.

[50] S. N. Chow, J. A. Yorke. Lyapunov theory and perturbation of stable and asymptotically stable systems. *Journal of Differential Equations*, vol. 15, no. 2, pp. 308–321, 1974. DOI: 10.1016/0022-0396(74)90082-5.

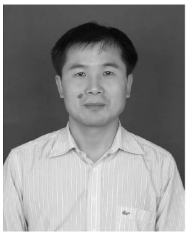
[51] J. Wang, N. Elia. Control approach to distributed optimization. In *Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing*, IEEE, Monticello, USA, pp. 557–561, 2010. DOI: 10.1109/ALLERTON.2010.5706956.

Hao-Ran Yang received the B. Sc. degree in statistics from Lingnan Normal University, China in 2015. He is currently a master student in Nanchang University, China.

His research interests include distributed optimization and multi-agent systems.

E-mail: 13410901483@163.com

ORCID iD: 0000-0001-5238-2723



Wei Ni received the Ph. D. degree in systems science from Academy of Mathematics and Systems Science, Chinese Academy of Sciences, China in 2010. He is currently an associate professor with School of Science, Nanchang University, China.

His research interests include control of switched and impulsive systems, and complex systems.

E-mail: niwei@ncu.edu.cn (Corresponding author)

ORCID iD: 0000-0002-1410-8317