



A projection-based continuous-time algorithm for distributed optimization over multi-agent systems

Xingnan Wen¹ · Sitian Qin¹

Received: 30 September 2020 / Accepted: 18 December 2020 / Published online: 11 January 2021
© The Author(s) 2021

Abstract

Multi-agent systems are widely studied due to its ability of solving complex tasks in many fields, especially in deep reinforcement learning. Recently, distributed optimization problem over multi-agent systems has drawn much attention because of its extensive applications. This paper presents a projection-based continuous-time algorithm for solving convex distributed optimization problem with equality and inequality constraints over multi-agent systems. The distinguishing feature of such problem lies in the fact that each agent with private local cost function and constraints can only communicate with its neighbors. All agents aim to cooperatively optimize a sum of local cost functions. By the aid of penalty method, the states of the proposed algorithm will enter equality constraint set in fixed time and ultimately converge to an optimal solution to the objective problem. In contrast to some existed approaches, the continuous-time algorithm has fewer state variables and the testification of the consensus is also involved in the proof of convergence. Ultimately, two simulations are given to show the viability of the algorithm.

Keywords Deep reinforcement learning · Multi-agent systems · Distributed optimization problem · Penalty method · Convergence

Introduction

Reinforcement learning stems from an experiment on the behaviors of cats in 1898 by Thorndike [20]. With the development of science and application, the concept of deep reinforcement learning (DRL) is proposed. DRL is an interdisciplinary of reinforcement learning and deep learning to cope with environments with high dimensions [17]. In DRL field, multi-agent systems raise much attention, since they own the ability to deal with complex tasks by the cooperation of individual agents. Thus, the study among multi-agent systems is important.

Owing to the widespread application in resource allocation [1], machine learning [21], and robot control [25], and so on, distributed optimization problem over multi-agent systems has drawn much attention in recent years. In resolving

process, agents can only communicate with their neighbors and the optimal solutions of distributed optimization problems are calculated based on their interaction. In past few decades, many research results have emerged for distributed optimization problems with sorts of constraints (see [3,14,18] and references therein). In [18], a topology network is introduced to solve a distributed optimization problem, where time delays are considered in the communication process among agents. By augmented Lagrange multiplier method, a nonsmooth distributed optimization problem with inequality constraints is studied in [29]. Moreover, a penalty method is utilized in [14] for distributed optimization to simplify constraints on an undirected topology graph.

Neurodynamic approach is a powerful tool in distributed optimization problems as well as traditional optimizations. For solving optimizations, Xia et al. [24] proposed a neurodynamic approach with global convergence. In [22], a second-order distribution dynamics is introduced to solve an optimization without constraints. In application of neurodynamic approaches, time delays are often inevitable. To consider the influence of time delays, Zeng et al. presented a continuous-time algorithm with time-varying delays and studied their stabilities in [31,32]. To be suitable for comput-

✉ Sitian Qin
qinsitian@163.com; qinsitian@hitwh.edu.cn
Xingnan Wen
hitwhwxn@163.com

¹ Department of Mathematics, Harbin Institute of Technology, Weihai 264209, People's Republic of China

ers, discrete-time recurrent algorithms are also considered in [30].

In recent years, for the purpose of realizing the goal of reducing the layers of related models, penalty method has attracted great attention among scholars. In [33], a discrete-time algorithm is proposed to solve distributed optimization problem without constraints. Authors in [33] discuss the convergence and error tolerance with the help of estimating a penalty parameter in advance. To solve distributed optimization problems with double local constraints, a neurodynamic approach is proposed in [16] by projection and regularization methods. It is worth noting that the computational burden may increase when applying neural network in [16] to solve distributed optimization problems with complex structures. For the broader application, Jiang et al. [8] proposed a penalty-like continuous-time algorithm for distributed optimization problems with affine equality and convex inequality constraints without considering penalty method.

It can be seen that distributed optimization problems are detailed studied with sorts of constraints and topology graphs according to above research findings. And it is worth noting that the computational complexity varies among different models. To introduce a proper algorithm to simply the calculation and complexity, the penalty method utilized in [14] is also considered here. Besides, we construct a projection-based continuous-time algorithm to solve distributed optimization problems with equality and inequality constraints over multi-agent systems in this paper. The main contributions of this paper are listed as follows:

1. In this paper, a continuous-time algorithm is proposed to resolve distributed optimization problem over multi-agent systems. Since the local cost functions and inequality constraint functions are studied without smoothness assumption, the distributed optimization problem here is more general than the one in [15]. Moreover, algorithm is constructed as a differential inclusion system, which is more flexible than the general gradient systems such as [6,26].
2. By introducing penalty method, the distributed optimization problem with equality and inequality constraints is transformed into a new one. Thus, the distributed optimization problem can be solved by the algorithm which can be applied into the reformulated problem. Compared with conventional approach for distributed optimization problems in [12], penalty method has the advantage of reducing the dimension of models.
3. From any initial point, the states of the continuous-time algorithm are sure to enter equality constraint in a fixed time. This property guarantees the algorithm to possess the advantage of simplifying the distributed optimization problems by ignoring one of the constraints after a period time. Since fixed entering time is obtained without calcu-

lating $(AA^T)^{-1}$, the continuous-time algorithm is easier to be applied than algorithm in [11].

4. Different from conventional approaches which rely on introducing Laplace matrix to guarantee the consensus, the continuous-time algorithm realizes the consensus in the process of dealing with penalty terms. Compared with algorithm in [12], approach herein has fewer state variables, which is more preferable in application fields.

The rest parts are organized as follows. In the section “Preliminaries and problem description”, some necessary preliminaries are first introduced. The targeted distributed optimization problem and the reformulated one are given in the section “Problem reformulation and neurodynamic approach”. In the section “Problem reformulation and neurodynamic approach”, we propose a continuous-time algorithm and discuss the convergence of its states. Double numerical examples are proposed in the section “Simulations” to verify the performance of the presented algorithm. In the section “Conclusions”, the conclusion of this paper and our future works are summarized.

Preliminaries and problem description

This section presents some basic concepts containing graph theory and nonsmooth analysis as well as some necessary lemmas. Then, our studied problem is introduced.

Graph theory and nonsmooth analysis

A graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$ is utilized in this paper to depict the information sharing among agents, where $\mathcal{N} = \{1, 2, \dots, m\}$ means the set of agents and $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ represents an edge set. An edge e_{ij} exists between a pair of agents (i, j) if they can communicate with each other. Denote that \mathcal{N}_i is the neighbor set of agent i , which is $\mathcal{N}_i = \{j \in \mathcal{N} : e_{ij} \in \mathcal{E}\}$. And a graph is called undirected if $(i, j) \in \mathcal{E}$ implies $(j, i) \in \mathcal{E}$. A path between agents i and j in graph \mathcal{G} is a sequence of edges $(i, i_1), (i_1, i_2), \dots, (i_k, j)$, where $i_1, i_2, \dots, i_k \in \mathcal{N}$. Undirected graph \mathcal{G} is said to be connected if there is a path between any pair of agents.

Definition 1 [4] *Let $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ be Lipschitz continuous near $u_0 \in \mathbb{R}^n$. The generalized directional derivative of Ψ at u_0 is defined as:*

$$\Psi^0(u_0; v) = \limsup_{u \rightarrow u_0, s \downarrow 0} \frac{\Psi(u + sv) - \Psi(u)}{s}.$$

Also, if $\Psi^0(u_0; v) = \Psi'(u_0; v)$, Ψ is said to be regular at u_0 , where:

$$\Psi'(u_0; v) = \lim_{s \downarrow 0} \frac{\Psi(u_0 + sv) - \Psi(u_0)}{t}.$$

Proposition 1 [4] Suppose the function $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex. Then, the subdifferential of Ψ at $u_0 \in \mathbb{R}^n$ is:

$$\partial\Psi(u_0) = \{\gamma \in \mathbb{R}^n : \langle \gamma, u - u_0 \rangle \leq \Psi(u) - \Psi(u_0), \forall u \in \mathbb{R}^n\}.$$

Proposition 2 [4] If $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ is Lipschitz continuous near $u_0 \in \mathbb{R}^n$ with Lipschitz constant α , then the following items hold:

- (1) $\partial\Psi(u_0)$ is upper semicontinuous at u_0 ;
- (2) $\partial\Psi(u_0)$ is a nonempty convex compact set;
- (3) $\|\xi\| \leq \alpha$ for any $\xi \in \partial\Psi(u_0)$.

Lemma 1 [4] If $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex on $\Omega \subseteq \mathbb{R}^n$, then for any $u_1, u_2 \in \Omega$, the subdifferential of Ψ satisfies:

$$\langle u_1 - u_2, \xi_1 - \xi_2 \rangle \geq 0,$$

for any $\xi_1 \in \partial\Psi(u_1)$ and $\xi_2 \in \partial\Psi(u_2)$.

Due to Proposition 2.3.6 in [4], a convex function $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ is also regular.

Lemma 2 [4] (Chain rule) Assume that $x(t) : \mathbb{R} \rightarrow \mathbb{R}^n$ is differentiable and Lipschitz, if function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ is regular at $x(t)$, then we have:

$$\dot{V}(x(t)) = \langle \xi, \dot{x}(t) \rangle, \forall \xi \in \partial V(x(t)),$$

for a.e. $t \in [0, +\infty)$.

Several necessary lemmas

Consider a system

$$\begin{cases} \dot{u}(t) = -\alpha u(t)^{2-\frac{a}{b}} - \beta u(t)^{\frac{a}{b}} \\ u(0) = u_0, \end{cases} \quad (1)$$

where $u(t), u_0 \in \mathbb{R}^n, \alpha, \beta$ are positive constants and $a, b > 0$ are odd integers satisfying $a < b$.

Definition 2 For any initial point $u(0) = u_0 \in \mathbb{R}^n$, if there exists a $T > 0$, such that:

- (1) $\lim_{t \rightarrow T} \|u(t) - u^*\| = 0$;
- (2) $\|u(t) - u^*\| = 0, t > T$,

where u^* is an equilibrium point of system (1). Then, the equilibrium point of system (1) is said to be fixed-time stable.

Lemma 3 [34] The original of system (1) is fixed-time stable and upper bound of the time can be calculated as:

$$T \leq \frac{\pi b}{2\sqrt{\alpha\beta}(b-a)}.$$

Lemma 4 [4] Suppose that $\Omega_1, \Omega_2 \subseteq \mathbb{R}^n$ are closed convex sets satisfying $\text{int}(\Omega_1 \cap \Omega_2) \neq \emptyset$, then for $u_0 \in \Omega_1 \cap \Omega_2$:

$$N_{\Omega_1 \cap \Omega_2}(u_0) = N_{\Omega_1}(u_0) + N_{\Omega_2}(u_0),$$

where $N_{\Omega}(u_0)$ is the normal cone to $\Omega \subseteq \mathbb{R}^n$ at u_0 , which is defined as $N_{\Omega}(u_0) = \{\xi \in \mathbb{R}^n | \langle \xi, u_0 - u \rangle \geq 0, \forall u \in \Omega\}$.

Lemma 5 [34] Suppose that $\Omega \subseteq \mathbb{R}^n$ is a nonempty closed convex set. Then, for any $u \in \mathbb{R}^n$, the projection operator:

$$\Pi_{\Omega}(u) = \arg \min_{v \in \Omega} \|v - u\|$$

has the following propositions:

- (1) $\|\Pi_{\Omega}(u) - u\|$ is continuous at u ;
- (2) $\nabla(\|\Pi_{\Omega}(u) - u\|^2) = -2(\Pi_{\Omega}(u) - u)$.

Distributed optimization problem

In this paper, we consider a network consisted of m agents over an undirected graph \mathcal{G} to cooperatively solve the distributed optimization problem as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^m f_i(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, \\ & A_i x = b_i, i = 1, 2, \dots, m, \end{aligned} \quad (2)$$

where $x \in \mathbb{R}^n, f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ are local cost functions and $g_i(x) = (g_{i1}(x), g_{i2}(x), \dots, g_{ip_i}(x))^T : \mathbb{R}^n \rightarrow \mathbb{R}^{p_i}$ may be nonsmooth, $A_i \in \mathbb{R}^{q \times n}$ are row full rank matrixes and $b_i \in \mathbb{R}^q$. For convenience, we denote:

$$\begin{aligned} S_i &= \{x \in \mathbb{R}^n | g_i(x) \leq 0\}, \\ \Omega_i &= \{x \in \mathbb{R}^n | A_i x = b_i\}, \end{aligned}$$

for $i = 1, 2, \dots, m$. In this paper, we default there exists at least one minimum of distributed optimization problem (2).

For solving distributed optimization problem (2), some useful assumptions are first introduced here.

Assumption 1 The graph \mathcal{G} is undirected and connected.

Assumption 2 The Slater condition of distributed optimization problem (2) holds, i.e., there exists a $\tilde{x} \in \mathbb{R}^n$, such that $g_i(\tilde{x}) < 0$ as well as $A_i \tilde{x} = b_i$ for $i = 1, 2, \dots, m$. Besides, the sets S_i are bounded; that is $\exists M_i > 0$, such that $S_i \subseteq B(\tilde{x}, M_i)$ for $i = 1, 2, \dots, m$.

Assumption 3 The local cost functions $f_i(x)$ are all convex and Lipschitz on \mathbb{R}^n with the Lipschitz constant L .

Remark 1 The Assumptions 1–3 are widely adistributed optimization problems in solving distributed optimization problems, such as [12,15]. Assumption 1 guarantees agents can communicate with their neighbors. In addition, Assumptions 2-3 imply that distributed optimization problem (2) is solvable.

Problem reformulation and neurodynamic approach

In this section, based on penalty method, the distributed optimization problem (2) is equivalently converted to a new one. Besides, a neurodynamic approach is proposed to solve the obtained distributed optimization problem.

Problem reformulation

Consider the following distributed optimization problem:

$$\begin{aligned} \min \quad & \sum_{i=1}^m f_i(x_i) + \sigma \sum_{i=1}^m G_i(x_i) + \lambda \sum_{i=1}^m \sum_{j \in \mathcal{N}_i} \|x_i - x_j\| \\ \text{s.t.} \quad & x_i \in \Omega_i, i = 1, 2, \dots, m, \end{aligned} \tag{3}$$

where f_i are from distributed optimization problem (2), $\Omega_i = \{x \in \mathbb{R}^n | A_i x_i = b_i\}$, the functions $G_i(x_i) = \sum_{k=1}^{p_i} \max\{g_{ik}(x_i), 0\}$, and $\sigma, \lambda > 0$ are penalty parameters.

Proposition 3 Let Assumption 2 holds, and then, the functions $G_i(x_i)$ in distributed optimization problem (3) are coercive, that is:

$$\lim_{\|x_i\| \rightarrow +\infty} G_i(x_i) = +\infty, \tag{4}$$

for $i = 1, 2, \dots, m$.

Let $\tilde{g} = -\max_{1 \leq i \leq m, 1 \leq k \leq p_i} \{g_{ik}(\tilde{x})\}$. Then, we have the following conclusion.

Theorem 1 Suppose that Assumptions 1–3 hold. If $\Omega_i = \Omega_j, i, j \in \{1, 2, \dots, m\}$, then there exist $\sigma_0 = \frac{ML}{m\tilde{g}}$ and $\lambda_0 = m(L + \sigma_0)$, such that distributed optimization problem (3) is equivalent to distributed optimization problem (2) with $\sigma > \sigma_0$ and $\lambda > \lambda_0$.

Proof Consider a distributed optimization problem as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^m f_i(x) + \sigma \sum_{i=1}^m G_i(x) \\ \text{s.t.} \quad & x \in \Omega_i, i = 1, 2, \dots, m; \end{aligned} \tag{5}$$

then, we will first prove distributed optimization problem (2) is equivalent to problem (5).

Suppose that x^* is an optimal solution to distributed optimization problem (2), which yields:

$$0 \in \sum_{i=1}^m \partial f_i(x^*) + N_{S \cap \Omega}(x^*), \tag{6}$$

where $S = \bigcap_{i=1}^m S_i$ and $\Omega = \bigcap_{i=1}^m \Omega_i$. According to Assumption 2 and Lemma 4, $\text{int}(S) \cap \Omega \neq \emptyset$ implies that:

$$\begin{aligned} N_{S \cap \Omega}(x^*) &= N_S(x^*) + N_{\Omega}(x^*) \\ &= \sum_{i=1}^m N_{S_i}(x^*) + \sum_{i=1}^m N_{\Omega_i}(x^*). \end{aligned} \tag{7}$$

Similar to [28], we have:

$$N_{S_i}(x^*) = \bigcup_{\mu \geq 0} \mu \partial G_i(x^*) = \sum_{k \in I_{i0}(x^*)} [0, +\infty) \partial g_{ik}(x^*), \tag{8}$$

where $I_{i0}(x^*) = \{k \in \{1, 2, \dots, p_i\} : g_{ik}(x^*) = 0\}$. Thus, combining (7) with (8), it can be obtained that:

$$N_{S \cap \Omega}(x^*) = \sum_{i=1}^m \sum_{k \in I_{i0}(x^*)} [0, +\infty) \partial g_{ik}(x^*) + \sum_{i=1}^m N_{\Omega_i}(x^*). \tag{9}$$

By (6), there exists a $\xi^* \in N_{S \cap \Omega}(x^*)$, such that $-\xi^* \in \sum_{i=1}^m \partial f_i(x^*)$. Then, it is obvious that $\|\xi^*\| \leq L$, where L is the Lipschitz constant of $f_i(x)$ given in Assumption 3.

Hence, there exist $\sigma_{ik} \in [0, +\infty), \varsigma_{ik}^* \in \partial g_{ik}(x^*)$ for $k \in I_{i0}(x^*)$ as well as $\tau_i^* \in N_{\Omega_i}(x^*)$, such that:

$$\xi^* = \sum_{i=1}^m \sum_{k \in I_{i0}(x^*)} \sigma_{ik} \varsigma_{ik}^* + \sum_{i=1}^m \tau_i^*. \tag{10}$$

Then, one has:

$$\sigma_0 \geq \sigma_{ik}, \tag{11}$$

for any $k \in I_{i_0}(x^*)$. If (11) does not hold, then there exists $\sigma_{ik_0} > \sigma_0, k_0 \in I_{i_0}(x^*)$. It follows that:

$$\begin{aligned} \langle \xi^*, x^* - \tilde{x} \rangle &= \sum_{i=1}^m \left\langle \sum_{k \in I_{i_0}(x^*)} \sigma_{ik} \varsigma_{ik}^* + \tau_i^*, x^* - \tilde{x} \right\rangle \\ &= \sum_{i=1}^m \sum_{k \in I_{i_0}(x^*)} \sigma_{ik} \langle \varsigma_{ik}^*, x^* - \tilde{x} \rangle \\ &\quad + \sum_{i=1}^m \langle \tau_i^*, x^* - \tilde{x} \rangle, \end{aligned} \tag{12}$$

where $\tilde{x} \in \Omega \cap \text{int}(S)$ is from Assumption 2. Since $\varsigma_{ik}^* \in \partial g_{ik}(x^*)$, then by Definition 1, one has:

$$g_{ik}(\tilde{x}) - g_{ik}(x^*) \geq \langle \varsigma_{ik}^*, \tilde{x} - x^* \rangle, \tag{13}$$

for any $k \in I_{i_0}(x^*), i \in \{1, 2, \dots, m\}$. Besides, $\tau_i^* \in N_{\Omega_i}(x^*)$ and Lemma 4 imply that:

$$\langle \tau_i^*, \tilde{x} - x^* \rangle \leq 0. \tag{14}$$

for any $i \in \{1, 2, \dots, m\}$. Substituting (13) and (14) into (12), it can be immediately obtained that:

$$\begin{aligned} \langle \xi^*, x^* - \tilde{x} \rangle &\geq - \sum_{i=1}^m \sum_{k \in I_{i_0}(x^*)} \sigma_{ik} g_{ik}(\tilde{x}) \\ &\geq \sum_{i=1}^m \sum_{k \in I_{i_0}(x^*)} \sigma_{ik} \tilde{g} \geq m \sigma_{ik_0} \tilde{g}, \end{aligned} \tag{15}$$

which together with $\sigma_{ik_0} > \sigma_0, k_0 \in I_{i_0}(x^*)$ and $\sigma_0 = \frac{ML}{m\tilde{g}}$ mean:

$$\langle \xi^*, x^* - \tilde{x} \rangle > ML.$$

It contradicts with the fact:

$$\langle \xi^*, x^* - \tilde{x} \rangle \leq \|\xi^*\| \cdot \|x^* - \tilde{x}\| \leq ML.$$

Thus, (11) holds and it delivers:

$$\begin{aligned} \xi^* &= \sum_{i=1}^m \sum_{k \in I_{i_0}(x^*)} \sigma_{ik} \varsigma_{ik}^* + \sum_{i=1}^m \tau_i^* \\ &\in \sum_{i=1}^m \sum_{k \in I_{i_0}(x^*)} \sigma_{ik} \partial g_{ik}(x^*) + \sum_{i=1}^m N_{\Omega_i}(x^*) \\ &\subseteq \sigma \sum_{i=1}^m \sum_{k \in I_{i_0}(x^*)} [0, 1] \partial g_{ik}(x^*) + N_{\Omega}(x^*) \\ &= \sigma \sum_{i=1}^m \partial G_i(x^*) + N_{\Omega}(x^*), \end{aligned} \tag{16}$$

for any $\sigma > \sigma_0$. Moreover, $-\xi^* \in \sum_{i=1}^m \partial f_i(x^*)$ and (16) show:

$$0 \in \sum_{i=1}^m \partial f_i(x^*) + \sigma \sum_{i=1}^m \partial G_i(x^*) + N_{\Omega}(x^*),$$

which implies that x^* is an optimal solution to distributed optimization problem (5). It is not difficult to prove that the minimum of distributed optimization problem (5) is also the optimal solution to problem (2). Hence, the equivalence of distributed optimization problem (2) and (5) has been testified.

On the other hand, by the proof of Theorem 1 in [33], if $\lambda > \lambda_0 = m(L + \sigma_0)$, the optimal solutions of distributed optimization problem (5) also minimize the cost function in (3) and vice versa. Therefore, the distributed optimization problem (2) is equivalent to problem (3) with $\sigma > \sigma_0$ and $\lambda > \lambda_0$. \square

Remark 2 Penalty method is a widely adopted approach in solving optimization problems with sorts of constraints, such as [19,28] and so on. By introducing penalty parameters, one can reduce the dimension of related algorithms. In this paper, we introduce double penalty parameters to reformulated original distributed optimization problem into a new one. Compared with the conventional continuous-time algorithm in [12], the model herein has the advantages of owning lower state variables.

According to Theorem 1, the equivalence of distributed optimization problem (2) and (3) are based on $\Omega_i = \Omega_j, i, j \in \{1, 2, \dots, m\}$. Let $\Omega_i = \Omega$, where:

$$\Omega \triangleq \{x \in \mathbb{R}^n | Ax_i = b\},$$

and $A_i = A \in \mathbb{R}^{m \times n}$ and $b_i = b \in \mathbb{R}^m$ for $i = 1, 2, \dots, m$. In the following part of this paper, we will default this condition.

Projection-based continuous-time algorithm

To solve distributed optimization problem (3), a projection-based continuous-time algorithm is proposed as follows:

$$\begin{aligned} \dot{x}_i(t) &\in |v_i(t)|^{2-\frac{a}{b}} \text{sign}(v_i(t)) + |v_i(t)|^{\frac{a}{b}} \text{sign}(v_i(t)) \\ &\quad - u_i(t) + |u_i(t)| \text{sign}(v_i(t)), \end{aligned} \tag{17}$$

where

$$u_i(t) = \partial f_i(x_i(t)) + \sigma \partial G_i(x_i(t)) + 2\lambda \sum_{j \in \mathcal{N}_i} \partial \|x_i(t) - x_j(t)\|$$

$$v_i(t) = \Pi_{\Omega}(x_i(t)) - x_i(t);$$

a, b are positive odds and satisfy $a < b, \partial f_i(\cdot), \partial G_i(\cdot)$ and $\partial \|\cdot\|$ are subdifferentials of corresponding functions, $\sigma, \lambda >$

0 are penalty parameters, and Π_Ω represent the projection operator on the sets $\Omega = \{x \in \mathbb{R}^n | Ax_i = b\}$. It is worth noticing $|u_i| = \{|\xi_i| : \xi_i \in u_i\}$ for $i = 1, 2, \dots, m$ in (17).

Remark 3 The continuous-time algorithm (17) is mainly composed of the following two parts:

(1): The terms $|v_i(t)|^{2-\frac{a}{b}} \text{sign}(v_i(t)) + |v_i(t)|^{\frac{a}{b}} \text{sign}(v_i(t))$ and $|u_i(t)| \text{sign}(v_i(t))$ are utilized to guarantee the states of continuous-time algorithm (17) will enter Ω , $i = 1, 2, \dots, m$ in fixed time.

(2): The terms $-\partial f_i(x_i(t)) - \sigma \partial G_i(x_i(t)) - 2\lambda \sum_{j \in \mathcal{N}_i} \partial \|x_i(t) - x_j(t)\|$ are the subdifferentials of objective function in distributed optimization problem (3). They can be regarded as a gradient descent method and ensure the states of continuous-time algorithm (17) converge to an optimal solution of distributed optimization problem (3).

The projection-based continuous-time algorithm (17) can be also expressed by the following form:

$$\dot{\mathbf{x}}(t) \in |\mathbf{v}(t)|^{2-\frac{a}{b}} \mathbf{sign}(\mathbf{v}(t)) + |\mathbf{v}(t)|^{\frac{a}{b}} \mathbf{sign}(\mathbf{v}(t)) - \mathbf{u}(t) + |\mathbf{u}(t)| \mathbf{sign}(\mathbf{v}(t)), \tag{18}$$

where $\mathbf{x} = \text{col}\{x_1, x_2, \dots, x_m\}$, $\mathbf{v} = \text{col}\{v_1, v_2, \dots, v_m\}$ and $\mathbf{u} = \text{col}\{u_1, u_2, \dots, u_m\}$. The continuous-time algorithm (17) is the distributed form of (18) and it can be seen that algorithm (17) is fully distributed.

Proposition 4 *The equilibrium point of continuous-time algorithm (18) is an optimal solution to distributed optimization problem (3) and vice versa.*

Convergence analysis

In this part, with the help of Lyapunov method and above preliminaries, we will study the convergence of continuous-time algorithm (17).

Theorem 2 *Let Assumptions 1–3 hold. For any initial point $\mathbf{x}(0) = \mathbf{x}^0 = \text{col}\{x_1^{(0)}, x_2^{(0)}, \dots, x_m^{(0)}\} \in \mathbb{R}^{mn}$, the state solution $\mathbf{x}(t)$ of neural network (18) will converge to $\Omega = \Omega_1 \times \Omega_2 \times \dots \times \Omega_m$ and stays in it in fixed time.*

Proof Suppose that $\mathbf{x}(t) = \text{col}\{x_1(t), x_2(t), \dots, x_m(t)\}$ is the solution with initial point $\mathbf{x}^{(0)} \in \mathbb{R}^n$. By the definition of subdifferential, there exist $\eta_i(t) \in \partial f_i(x_i(t))$, $\gamma_i(t) \in \partial G_i(x_i(t))$, and $\theta_i(t) \in \sum_{j \in \mathcal{N}_i} \partial \|x_i(t) - x_j(t)\|$, such that:

$$\xi_i(t) = \eta_i(t) + \sigma \gamma_i(t) + 2\lambda \theta_i(t), \tag{19}$$

for a.e. $t \geq 0$, where $\xi_i(t) \in u_i(t)$ is a vector.

Consider the Lyapunov function:

$$V_i(x_i) = \frac{1}{2}(\Pi_\Omega(x_i) - x_i)^2.$$

According to Lemma 5 and (19), we can take the derivative of $V_i(x)$ along $x_i(t)$ as follows:

$$\begin{aligned} \frac{d}{dt} V_i(x_i(t)) &= -\langle \Pi_\Omega(x_i(t)) - x_i(t), \dot{x}_i(t) \rangle \\ &= -\langle v_i(t), -\xi_i(t) + |\xi_i(t)| \text{sign}(v_i(t)) \rangle \\ &\quad - |v_i(t)|^{3-\frac{a}{b}} - |v_i(t)|^{1+\frac{a}{b}}, \end{aligned} \tag{20}$$

for a.e. $t \geq 0$.

Since $\langle v_i(t), \xi_i(t) \rangle - |\xi_i(t)| |v_i(t)| \leq 0$, then it follows from (20) that:

$$\begin{aligned} \frac{d}{dt} V_i(x_i(t)) &\leq -|v_i(t)|^{3-\frac{a}{b}} - |v_i(t)|^{1+\frac{a}{b}} \\ &= -(v_i^2(t))^{\frac{3b-a}{2b}} - (v_i^2(t))^{\frac{a+b}{2b}} \\ &= -(2V_i(x_i(t)))^{\frac{3b-a}{2b}} - (2V_i(x_i(t)))^{\frac{a+b}{2b}}. \end{aligned} \tag{21}$$

Let $\rho_i(x_i) = \sqrt{2V_i(x_i)}$, and then, it can be immediately obtained that:

$$\frac{d}{dt} \rho_i(x_i(t)) \leq -(\rho_i(x_i(t)))^{2-\frac{a}{b}} - (\rho_i(x_i(t)))^{\frac{a}{b}}. \tag{22}$$

By Lemma 3, we have $\lim_{t \rightarrow T} \rho_i(x_i(t)) = 0$, which also implies $\lim_{t \rightarrow T} V_i(x_i(t)) = 0$ for $i = 1, 2, \dots, m$. Thus, we have:

$$\Pi_\Omega(x_i^*) = x_i^*,$$

for $i = 1, 2, \dots, m$, which means for any initial point $\mathbf{x}^0 = \text{col}\{x_1^{(0)}, x_2^{(0)}, \dots, x_m^{(0)}\} \in \mathbb{R}^{mn}$, $\mathbf{x}(t)$ will enter the set in fixed time and T is bounded by:

$$T \leq \frac{b\pi}{2(b-a)}.$$

□

Remark 4 It is worth noting that the property of entering one of the constraints or feasible region is possessed by many continuous-time algorithms for solving optimization problems, such as [8, 11, 19, 34]. This proposition guarantees the continuous-time algorithm (17) which has the advantages of simplifying the distributed optimization problems by ignoring one of the constraints once the states enter this set. Compared with [11], the continuous-time algorithm (17) in constructed and fixed entering time is obtained without calculating $(AA^T)^{-1}$, which is easier to be applied in engineering fields.

Lemma 6 [2] (Opial) *Suppose that $x : [0, +\infty)$ is a curve. Then, $x(t)$ will be convergent to a point in $C \subseteq \mathbb{R}^n$ ($C \neq \emptyset$) as $t \rightarrow +\infty$ if and only if the following two tips are satisfied*

- (1) All accumulation points of $x(\cdot)$ are in C ;
- (2) For any $x^* \in C$, $\lim_{t \rightarrow +\infty} \|x(t) - x^*\|$ exists.

Theorem 3 Assume that Assumptions 1–3 hold, then for any initial point $\mathbf{x}(0) = \mathbf{x}^0 = \text{col}\{x_1^{(0)}, x_2^{(0)}, \dots, x_m^{(0)}\} \in \mathbb{R}^{mn}$, the state solution $\mathbf{x}(t)$ of neural network (17) will converge to its equilibrium point.

Proof Suppose $\mathbf{x}^* = \text{col}\{x_1^*, x_2^*, \dots, x_m^*\}$ and C^* are an equilibrium point and equilibrium point set of neural network (18). According to Theorem 2, from initial point $\mathbf{x}^0 = \text{col}\{x_1^{(0)}, x_2^{(0)}, \dots, x_m^{(0)}\} \in \mathbb{R}^{mn}$, the state solution $\mathbf{x}(t)$ of neural network (17) will enter Ω in fixed time. And the fixed time $T \leq \frac{b\pi}{2(b-a)}$. In other words, we can immediately get:

$$x_i(t) = \Pi_\Omega(x_i(t)),$$

for $t \geq T$. Hence, we have $v_i(t) = 0, t \geq T$.

Then, the neural network can be simplified as:

$$\dot{x}_i(t) \in -u_i(t), \tag{23}$$

for $t \geq T$, where $u_i(t)$ is from (17). In the following part, we will study the convergence of continuous-time algorithm (23).

Consider the Lyapunov function:

$$W(t) = \frac{1}{2} \sum_{i=1}^m \|x_i(t) - x_i^*\|^2, \tag{24}$$

on $t \in [T, +\infty)$. Taking the derivative of $W(t)$, there exist $\eta_i(t) \in \partial f_i(x_i(t)), \gamma_i(t) \in \partial G_i(x_i(t))$ and $\theta_i(t) \in \sum_{j \in \mathcal{N}_i} \partial \|x_i(t) - x_j(t)\|$, such that:

$$\begin{aligned} \frac{d}{dt} W(t) &= \sum_{i=1}^m \langle x_i(t) - x_i^*, \dot{x}_i(t) \rangle \\ &= - \sum_{i=1}^m \langle x_i(t) - x_i^*, \eta_i(t) \rangle - \sigma \sum_{i=1}^m \langle x_i(t) - x_i^*, \gamma_i(t) \rangle - 2\lambda \sum_{i=1}^m \sum_{j \in \mathcal{N}_i} \langle x_i(t) - x_i^*, \theta_i(t) \rangle \end{aligned} \tag{25}$$

for a.e. $t \geq T$.

Since \mathbf{x}^* is an equilibrium point of algorithm (18), it is also an equilibrium point of algorithm (23). Thus, there exist $\eta_i^* \in \partial f_i(x_i^*), \gamma_i^* \in \partial G_i(x_i^*),$ and $\theta_i^* \in \sum_{j \in \mathcal{N}_i} \partial \|x_i^* - x_j^*\|$, such that:

$$0 = \eta_i^* + \sigma \gamma_i^* + 2\lambda \theta_i^*, \tag{26}$$

for $i = 1, 2, \dots, m$, which means that:

$$0 = \sum_{i=1}^m \langle x_i(t) - x_i^*, \eta_i^* + \sigma \gamma_i^* + 2\lambda \theta_i^* \rangle. \tag{27}$$

Combining (25) and (27), we have:

$$\begin{aligned} \frac{d}{dt} W(t) &= - \sum_{i=1}^m \langle x_i(t) - x_i^*, \eta_i(t) - \eta_i^* \rangle \\ &\quad - \sigma \sum_{i=1}^m \langle x_i(t) - x_i^*, \gamma_i(t) - \gamma_i^* \rangle \\ &\quad - 2\lambda \sum_{i=1}^m \sum_{j \in \mathcal{N}_i} \langle x_i(t) - x_i^*, \theta_i(t) - \theta_i^* \rangle. \end{aligned} \tag{28}$$

Due to the fact that $f_i(x_i), G_i(x_i),$ and $\sum_{j \in \mathcal{N}_i} \partial \|x_i - x_j\|$ are convex functions, then by Lemma 1, it follows:

$$\begin{aligned} \langle x_i(t) - x_i^*, \eta_i(t) - \eta_i^* \rangle &\geq 0, \\ \langle x_i(t) - x_i^*, \gamma_i(t) - \gamma_i^* \rangle &\geq 0, \\ \sum_{j \in \mathcal{N}_i} \langle x_i(t) - x_i^*, \theta_i(t) - \theta_i^* \rangle &\geq 0, \end{aligned}$$

for a.e. $t \geq T$ and $i = 1, 2, \dots, m$.

Hence, it can be immediately obtained by (28) that:

$$\frac{d}{dt} W(t) \leq 0,$$

for a.e. $t \geq T$. Because of the nonnegativeness of $\sum_{i=1}^m \|x_i(t) - x_i^*\|^2$, we can declare that $\lim_{t \rightarrow +\infty} W(t)$ exists.

On the other hand, for the convenience of discussion, let:

$$K(\mathbf{x}) = \sum_{i=1}^m \left(f_i(x_i) + \sigma G_i(x_i) + \lambda \sum_{j \in \mathcal{N}_i} \|x_i - x_j\| \right), \tag{29}$$

where $\mathbf{x} = \text{col}\{x_1, x_2, \dots, x_m\}$ and $\Omega = \Omega_1 \times \Omega_2 \times \dots \times \Omega_m$. It is clear that $\liminf_{t \rightarrow +\infty} K(\mathbf{x}(t)) \geq \min_{\mathbf{x} \in \Omega} \{K(\mathbf{x})\}$. If $\liminf_{t \rightarrow +\infty} K(\mathbf{x}(t)) > \min_{\mathbf{x} \in \Omega} \{K(\mathbf{x})\}$, which means that, for any $\varepsilon > 0$, there exists $\tilde{T} > T$, such that:

$$K(\mathbf{x}(t)) > \min_{\mathbf{x} \in \Omega} \{K(\mathbf{x})\} + \varepsilon. \tag{30}$$

Since $\mathbf{x}^* = \text{col}\{x_1^*, x_2^*, \dots, x_m^*\}$ is an equilibrium point of continuous-time algorithm (18), by Proposition 4, \mathbf{x}^* is also an optimal solution to distributed optimization problem (3). It is obvious that:

$$K(\mathbf{x}^*) = \min_{\mathbf{x} \in \Omega} \{K(\mathbf{x})\}.$$

Table 1 Comparison of algorithms for solving distributed optimization problems

References	Objective functions	Constraint	Penalty parameter
[15]	Convex, differentiable	Unconstrained	Yes
[5]	Convex, nonsmooth	Unconstrained	No
[13]	Convex, nonsmooth	Bounded constraint	No
[12]	Convex, nonsmooth	Equality constraint inequality constraint	No
Herein	Convex, nonsmooth	Equality constraint inequality constraint	Yes

For the reason that $x_i(t) \in \Omega$ for $t \geq T$, then $x(t) \in \Omega$ for $t \geq \tilde{T} > T$. By (25) and (30), we have:

$$\frac{d}{dt}W(t) \leq \min_{x \in \Omega}\{K(x)\} - K(x(t)) \leq -\varepsilon \tag{31}$$

for $t \geq \tilde{T}$. The first inequality holds by Definition 1, which is utilized to demonstrate that:

$$\begin{aligned} \langle x_i - x_i^*, \eta_i \rangle &\leq f_i(x_i^*) - f_i(x_i), \\ \langle x_i - x_i^*, \gamma_i \rangle &\leq G_i(x_i^*) - G_i(x_i), \\ 2 \sum_{j \in \mathcal{N}_i} \langle x_i - x_i^*, \theta_i \rangle &\leq \sum_{j \in \mathcal{N}_i} (\|x_i^* - x_j^*\| - \|x_i - x_j\|), \end{aligned}$$

where $\eta_i \in \partial f_i(x_i)$, $\gamma_i \in \partial G_i(x_i)$, and $\theta_i \in \sum_{j \in \mathcal{N}_i} \partial \|x_i - x_j\|$. By calculating integral of (31) from \tilde{T} to $+\infty$, we have:

$$\lim_{t \rightarrow +\infty} \left(\sum_{i=1}^m \|x_i(t) - x_i^*\|^2 \right) = -\infty, \tag{32}$$

which leads to a contradiction. Hence, we obtain $\liminf_{t \rightarrow +\infty} K(x(t)) = \min_{x \in \Omega}\{K(x)\}$.

Thus, there exists a time sequence $\{t_n\}$, such that:

$$\lim_{n \rightarrow +\infty} K(x(t_n)) = \min_{x \in \Omega}\{K(x)\},$$

which delivers by (31) that $W(t_n) \rightarrow 0$ as $n \rightarrow +\infty$. Since $\lim_{t \rightarrow +\infty} W(t)$ exists, we can get:

$$\lim_{t \rightarrow +\infty} W(t) = \lim_{t \rightarrow +\infty} \left(\frac{1}{2} \sum_{i=1}^m \|x_i(t) - x_i^*\|^2 \right) = 0.$$

Therefore, $x^* \in C^*$ is an accumulation of $x(t)$. By Lemma 6, it can be concluded that $x(t)$ of neural network (18) will converge to its equilibrium point. \square

According to Proposition 4, the equilibrium point of neural neural network (17) is an optimal solution to distributed optimization problem (2). Then, we have the following conclusion:

Corollary 1 Assume that Assumptions 1–3 hold, then for any initial point $x^0 = \text{col}\{x_1^{(0)}, x_2^{(0)}, \dots, x_m^{(0)}\} \in \mathbb{R}^n$ and $\sigma > \sigma_0, \lambda > \lambda_0$, then the states $x(t)$ of continuous-time algorithm (18) will converge to an optimal solution to distributed optimization problem (2).

Remark 5 In [12], distributed optimization problem (2) has been studied under the similar assumptions in this paper. With the help of Laplacian matrix, authors in [12] proposed a differential inclusion system and proved the consensus and convergence of agents. Different from [12], the continuous-time algorithm (17) achieves convergence by a transformed distributed optimization problem. Compared with algorithm in [12], continuous-time algorithm (17) has fewer state variables, which is more preferable in application fields. Here, we also introduce some algorithms to solve distributed optimization problems in Table 1 to make detailed comparisons.

Simulations

This section presents two numerical examples to show the viability of the conclusion above and the projection-based continuous-time algorithm. Details are given as follows:

Example 1 A three-agent-system communicating over an undirected connected graph is considered in this example and its interaction topology is presented in Fig. 1. The distributed

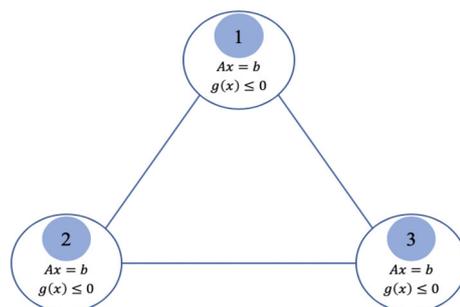


Fig. 1 Interaction topology of the three-agent system in Example 1

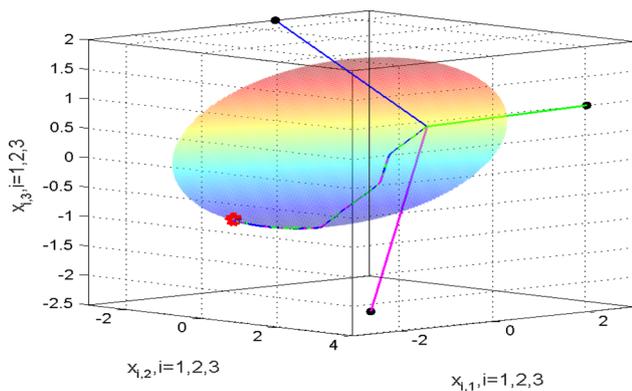


Fig. 2 Trajectories $(x_{i,1}, x_{i,2}, x_{i,3})$ of the states of continuous-time algorithm (17)

optimization problem is depicted as:

$$\begin{aligned} \min \quad & \sum_{i=1}^3 f_i(x_i) \\ \text{s.t.} \quad & x_{i,1} - 2x_{i,2} + x_{i,3} = 1 \\ & x_{i,1}^2 + x_{i,2}^2 + x_{i,3}^2 - 8 \leq 0, \end{aligned} \tag{33}$$

where $x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T \in \mathbb{R}^3$ is the decision variable of agent i . And the local cost function of distributed optimization problem (33) is defined as:

$$f_i(x_i) = M^T x_i + q, \quad i = 1, 2, 3, \tag{34}$$

where $q = 2$ is a constant and $M = (1, 3, 4)^T$. The equality constraint set is $\Omega = \{x_i \in \mathbb{R}^3 : x_{i,1} - 2x_{i,2} + x_{i,3} = 1\}$ to be specific. It is obvious that $A = (1, -2, 3)$ is a full-row-rank matrix and $b = 1$. In addition, the inequality constraint functions $g_i(x_i) = x_{i,1}^2 + x_{i,2}^2 + x_{i,3}^2 - 8, i = 1, 2, 3$. The sets $S_i = \{x_i : g_i(x_i) \leq 0\}$ are nonempty and satisfy that

$\text{int}(\Omega) \cap S_i \neq \emptyset$. Also, as the gradient of $f_i(x_i)$ can be calculated as $\nabla f_i(x_i) = M = [1, 3, 4]^T, i = 1, 2, 3$, apparently, it is bounded on the equality constraint set. Thus, all assumptions in this paper are satisfied.

Simulation results are presented in Figs. 2 and 3, which shows the trajectories of $(x_{i,1}, x_{i,2}, x_{i,3})^T$ of states with random initial values $x_1^{(0)} = (2, 1, -1)^T, x_2^{(0)} = (4, -3, 2)^T$ and $x_3^{(0)} = (1, 2, -2.4)^T$. The colored area in Fig. 2 is the feasible region of distributed optimization problem (33) and the above-mentioned initial points are marked as black ones. Besides, the star in red means an optimal solution of distributed optimization problem (33). It is easy to find that the state solutions of continuous-time algorithm (17) enter the equality constraint in finite time and stay in it thereafter. After that, the trajectories of all the three agents converge to the equilibrium point $(-0.3, -2.5, -1.233)^T$. The transient behaviors of the three agents' solutions of continuous-time algorithm (17) are also presented in Fig. 3.

Remark 6 In [9], Jia et al. proposed a generalized neural network for solving distributed optimization problems with inequality constraints only. Thus, the model cannot be immediately applied into distributed optimization problem (33) in Example 1. On the other hand, the continuous-time algorithm (17) is also suitable for distributed optimization problems in [9] under the condition that there is no equality constraint of distributed optimization problem (2) or the equality constraints are satisfied after the fixed time given in Theorem 2. Therefore, compared with neural network in [9], continuous-time algorithm (17) is more flexible when solving different distributed optimization problems.

Example 2 Consider a six-agent network with the communication topology shown in Fig. 4. And the distributed optimization problem is given as follows:

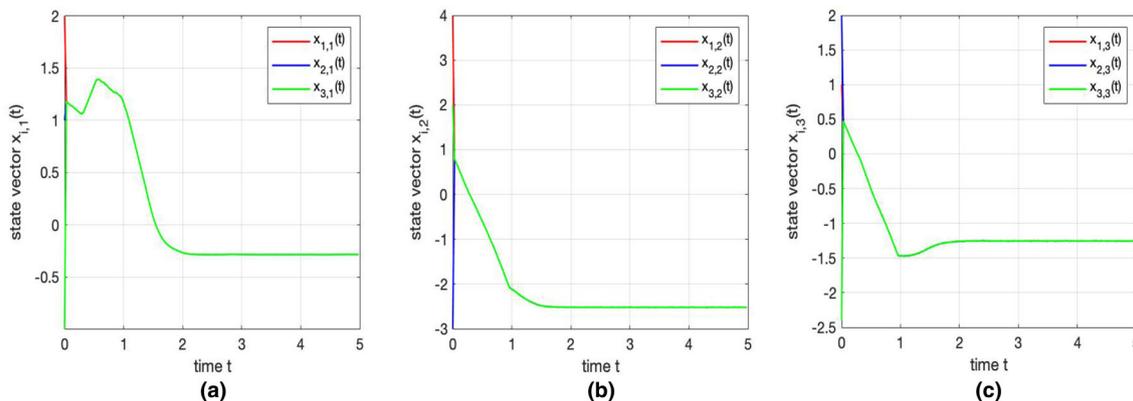


Fig. 3 Transient behaviors of agents' state solutions of continuous-time algorithm (17) in Example 1

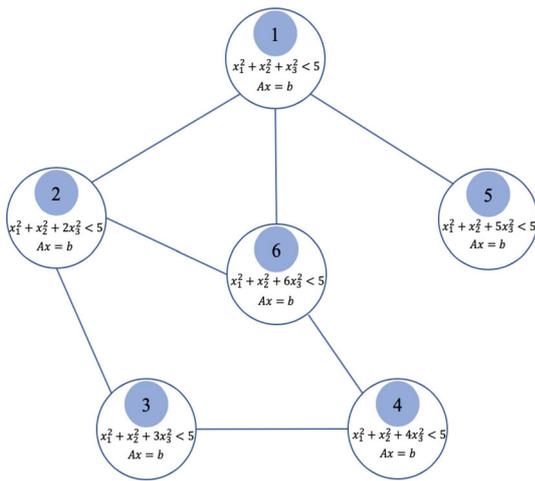


Fig. 4 Interaction topology of the six-agent system in Example 2

$$\begin{aligned} \min \quad & \sum_{i=1}^6 f_i(x_i) \\ \text{s.t.} \quad & x_{i,1} + x_{i,2} - 2x_{i,3} = 3, \\ & x_{i,1}^2 + x_{i,2}^2 + ix_{i,3}^2 - 5 \leq 0. \end{aligned} \tag{35}$$

The local cost functions of distributed optimization problem (35) are given as:

$$f_i(x_i) = \mathcal{M}_i^T x_i + q, \quad i = 1, 2, 3, 4, 5, 6, \tag{36}$$

where $\mathcal{M}_i = (1, 3, i)^T$ is relevant to each agent. Calculating the gradient of $f_i(x_i)$, one derives $\nabla f_i(x_i) = \mathcal{M}_i$. Despite the variability of the local cost functions, the boundedness of its gradient over the equality constraint set can still be guaranteed with a limited number of agents. It can be seen that $A = (1, 1, -2)$ is a full-row-rank matrix and:

$$g_i(x_i) = x_{i,1}^2 + x_{i,2}^2 + ix_{i,3}^2 - 5.$$

Apparently, distributed optimization problem (35) meets the requirements of the conditions assumed in this paper, for instance, the boundedness of the inequality constraint set and the nonemptiness of the feasible region.

The transient behaviors of the states of continuous-time algorithm (17) are presented in detail in Fig. 5, in which $x_{i,j}(t)$ represents the trajectories of states. Starting from initial points $x_1^{(0)} = (1.2, -2, -2.2)^T$, $x_2^{(0)} = (0.8, -2.5, -2.4)^T$, $x_3^{(0)} = (0.5, -2.8, -2.6)^T$, $x_4^{(0)} = (-0.5, -3, -2.8)^T$, $x_5^{(0)} = (-0.2, -3.3, -3.2)^T$, and $x_6^{(0)} = (-0.7, -3.5, -3.5)^T$, the states of continuous-time algorithm (17) converge to an optimal solution $\tilde{x} = (-1.447, -2.509, -3.478)^T$ of (35). The detailed movements of the three components of the decision variables are also presented in subplots of Fig. 5.

Conclusions

In this paper, a projection-based continuous-time algorithm is proposed to solve distributed optimization problems with equality and inequality constraints over multi-agent systems. By exact penalty method, the distributed optimization problem is reformulated to a new one without inequality constraints and consensus constraints. It is proved that from any initial points, the states of continuous-time algorithm will enter the equality constraint set of the transformed distributed optimization problem in fixed time. And states of continuous-time algorithm are proved to be convergent to an optimal solution of the original distributed optimization problem. Compared with existed models and approaches, the continuous-time algorithm has the advantages of owning fewer state variables. Besides, the states of continuous-time algorithm can find an optimal solution of distributed optimization problem under mild assumptions. Since the penalty method has the weakness of calculating penalty parameters, the application of continuous-time algorithm may be limited

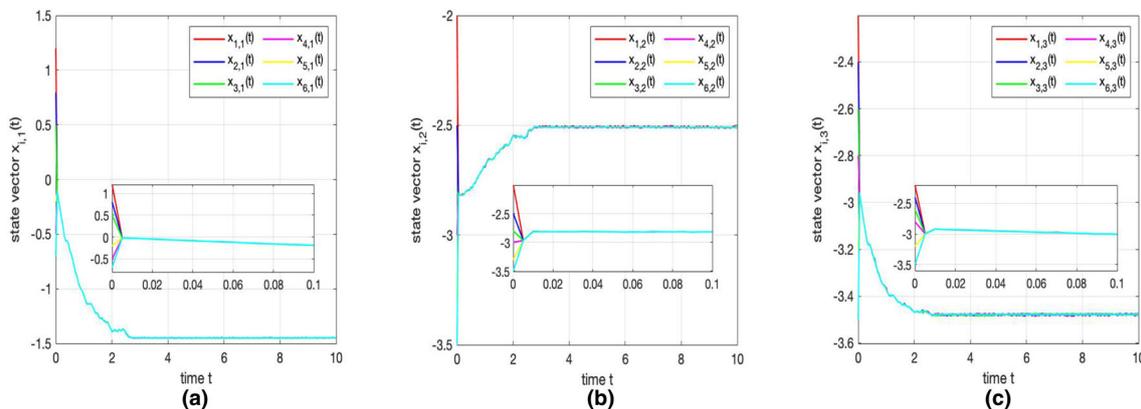


Fig. 5 Transient behaviors of the states of continuous-time algorithm (17) of the six agents in Example 2

in solving more complex distributed optimization problems. In our future work, we will consider to construct an algorithm independent of penalty method.

Acknowledgements This research was supported by National Natural Science Foundation of China (Grant 61773136) and National Natural Science Foundation of China (Grant 11471088).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Beck A, Nedic A, Ozdaglar A, Teboulle M (2014) Optimal distributed gradient methods for network resource allocation problems. *IEEE Trans Control Netw Syst* 1(1):64–74
- Bolte J (2003) Continuous gradient projection method in Hilbert spaces. *J Optim Theory Appl* 119(2):235–259
- Cherukuri A, Cortés J (2016) Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment. *Automatica* 74:183–193
- Clarke FH (1990) Optimization and nonsmooth analysis. *Siam* 5
- Gharesifard B, Cortes J (2014) Distributed continuous-time convex optimization on weight-balanced digraphs. *IEEE Trans Autom Control* 59(3):781–786
- Hu X, Wang J (2006) Solving pseudomonotone variational inequalities and pseudoconvex optimization problems using the projection neural network. *IEEE Trans Neural Netw* 17(6):1487–1499
- Hopfield J (1985) Neural computation of decisions in optimization problems. *Biol Cybern* 52
- Jiang X, Qin S, Xue X (2019) A penalty-like neurodynamic approach to constrained nonsmooth distributed convex optimization. *Neurocomputing*
- Jia W, Qin S, Xue X (2019) A generalized neural network for distributed nonsmooth optimization with inequality constraint. *Neural Netw*
- Kennedy M, Chua L (1988) Neural networks for nonlinear programming. *IEEE Trans Circuits Syst* 35(5):554–562
- Liu N, Qin S (2018) A neurodynamic approach to nonlinear optimization problems with affine equality and convex inequality constraints. *Neural Netw* 109
- Liu Q, Yang S, Wang J (2017) A collective neurodynamic approach to distributed constrained optimization. *IEEE Trans Neural Netw Learn Syst* 28(8):1747–1758
- Liu Q, Wang J (2015) A second-order multi-agent network for boundconstrained distributed optimization. *IEEE Trans Autom Control* 60(12):3310–3315
- Liang S, Zeng X, Hong Y (2018) Distributed nonsmooth optimization with coupled inequality constraints via modified Lagrangian function. *IEEE Trans Autom Control* 63(6):1753–1759
- Lin P, Ren W, Farrell JA (2016) Distributed continuous-time optimization: Nonuniform gradient gains, finite-time convergence, and convex constraint set. *IEEE Trans Autom Control*
- Ma L, Bian W (2019) A novel multiagent neurodynamic approach to constrained distributed convex optimization. *IEEE Trans Cybern* 1–12
- Nguyen T, Nguyen N, Nahavandi S (2020) Deep reinforcement learning for multiagent systems: a review of challenges, solutions, and applications. *IEEE Trans Cybern* 99:1–14
- Nedic A, Ozdaglar A (2009) Distributed subgradient methods for multi-agent optimization. *IEEE Trans Autom Control* 54(1):48–61
- Qin S, Yang X, Xue X, Song J (2017) A one-layer recurrent neural network for pseudoconvex optimization problems with equality and inequality constraints. *IEEE Trans Cybern* 47(10):3063–3074
- Thorndike E (1898) Animal intelligence: An experimental study of the associate processes in animals. *Am Psychol* 53(10):1125–1127
- Wang X, Han M (2014) Online sequential extreme learning machine with kernels for nonstationary time series prediction. *Neurocomputing* 145:90–97
- Wang J, Elia N (2011) Control approach to distributed optimization. *Commun Control Comput*
- Wang J, Liu Q (2015) A second-order multi-agent network for bound-constrained distributed optimization. *IEEE Trans Autom Control* 60(12):1–1
- Xia Y, Wang J (1998) A general methodology for designing globally convergent optimization neural networks. *IEEE Trans Neural Netw* 9(6):1331–1343
- Xia Y, Wang J (2001) A dual neural network for kinematic control of redundant robot manipulators. *IEEE Trans Syst Man Cybern* 31(1):147–154
- Xia Y, Wang J (2004) A general projection neural network for solving monotone variational inequalities and related optimization problems. *IEEE Trans Neural Netw* 15(2):318–328
- Xi C, Wu Q, Khan UA (2015) On the distributed optimization over directed networks. *Mathematics*,
- Xue X, Wei B (2008) Subgradient-based neural networks for nonsmooth convex optimization problems. *IEEE Trans Circuits Syst I Regul Pap* 55(8):2378–2391
- Yi P, Hong Y, Liu F (2015) Distributed gradient algorithm for constrained optimization with application to load sharing in power systems. *Syst Control Lett* 83:45–52
- Zeng Z, Wang J (2008) Design and analysis of high-capacity associative memories based on a class of discrete-time recurrent neural networks. *IEEE Trans Syst Man Cybern* 38(6):1525–1536
- Zeng Z, Wang J, Liao (2003) Global exponential stability of a general class of recurrent neural networks with time-varying delays. *IEEE Trans Circuits Syst I Fund Theory Appl* 50(10):1353–1358
- Zeng Z, Wang J (2006) Improved conditions for global exponential stability of recurrent neural networks with time-varying delays. *IEEE Trans Neural Netw* 17(3):623–635
- Zhang J, Life YK (2019) Distributed discrete-time optimization in multi-agent networks using only sign of relative state. *IEEE Trans Autom Control* 6(64):2352–2367
- Zuo Z, Lin T (2014) A new class of finite-time nonlinear consensus protocols for multi-agent systems. *Int J Control* 87(2):363–370

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.