KYBERNETIKA — VOLUME *59* (2023), NUMBER 3, PAGES 418–436

DISTRIBUTED ACCELERATED NASH EQUILIBRIUM LEARNING FOR TWO-SUBNETWORK ZERO-SUM GAME WITH BILINEAR COUPLING

Xianlin Zeng, Lihua Dou and Jinqiang Cui

This paper proposes a distributed accelerated first-order continuous-time algorithm for $O(1/t^2)$ convergence to Nash equilibria in a class of two-subnetwork zero-sum games with bilinear couplings. First-order methods, which only use subgradients of functions, are frequently used in distributed/parallel algorithms for solving large-scale and big-data problems due to their simple structures. However, in the worst cases, first-order methods for two-subnetwork zero-sum games often have an asymptotic or $O(1/t)$ convergence. In contrast to existing time-invariant first-order methods, this paper designs a distributed accelerated algorithm by combining saddle-point dynamics and time-varying derivative feedback techniques. If the parameters of the proposed algorithm are suitable, the algorithm owns $O(1/t^2)$ convergence in terms of the duality gap function without any uniform or strong convexity requirement. Numerical simulations show the efficacy of the algorithm.

Keywords: two-subnetwork zero-sum game, distributed accelerated algorithm, Nash equilibrium learning, nonsmooth function, continuous-time algorithm

Classification: 91A10, 37N40, 93A14

1. INTRODUCTION

Two-subnetwork zero-sum games are an important class of distributed non-cooperative games that have found a wide range of applications, including signal/image processing [4], statistical learning [29, 35, 40], formation control [21, 25, 44], and resource allocation [11, 20, 23, 41]. In a two-subnetwork zero-sum game (see [24, 34]), agents in one subnetwork cooperate to minimize their payoff function using local information exchange, and agents in the other subnetwork try to maximize the same payoff function. Two-network zero-sum games are powerful for modeling distributed robust optimization, distributed optimization with adversaries, and distributed optimization using the Lagrangian method. Hence, developing efficient distributed algorithms for solving a Nash equilibrium to two-network zero-sum games is very important.

Researchers have paid much attention to the research on distributed algorithms for solving a Nash equilibrium of two-subnetwork zero-sum games over multi-agent systems. Focusing on two-subnetwork zero-sum games, [15] proposed distributed algorithms using

DOI: 10.14736/kyb-2023-3-0418

time-invariant saddle-point dynamics over directed and undirected network topologies. To relax the requirement on fixed communication graphs, [24] proposed a subgradient-based distributed algorithm to compute a Nash equilibrium under uniformly jointly strongly connected directed graphs with homogeneous stepsizes. Focusing on nonsmooth cost functions and bounded constraints, [45] proposed a distributed continuous-time algorithm for min-max optimization (equivalent to two-network zero-sum games) using the proportional-integral design. Using the no-regret learning technique, [19] proposed a distributed mirror descent algorithm for computing a Nash equilibrium and established regret bounds on iterates using diminishing stepsizes and constant stepsizes. However, the prior research only focused on proving the convergence of algorithms to a Nash equilibrium and did not address estimating the convergence rate.

Recently, there is a substantial body of research on accelerated algorithms for centralized saddle point problems, which model classic zero-sum games. Focusing on saddle point problems, (time-invariant) saddle-point dynamics have possessed $O(1/t)$ or asymptotic convergence rates under the worst choice of convex/concave cost functions [12, 16]. By using adaptive parameters to increase the convergence rate, [6] has proposed a primal-dual algorithm with a rate of convergence $O(1/t^2)$ for saddle point problems under the assumption that either primal or dual cost function is uniformly convex. [43] has proposed an accelerated design for the primal-dual block coordinate method that has $O(1/t^2)$ convergence for the Lagrangian method with strongly convex functions. [42] has proposed a linearized augmented Lagrangian method with an acceleration $O(1/t^2)$ for strongly convex functions with adaptive parameters. [1] has proposed accelerated methods for solving smooth convex-concave saddle-point problems with a structure. [7] has focused on saddle point problems with bilinear couplings and proposed an accelerated method using a multistep acceleration scheme. All these works focus on centralized saddle point problems, and the analysis is hard.

Continuous-time optimization algorithms have been appealing for analyzing accelerated and distributed algorithms, because their analysis is often easier than that of discrete-time counterparts. When considering unconstrained convex optimization problems, works [26, 27] have shown that the best rate of convergence for first-order algorithms is $O(1/t^2)$ under convex cost functions. However, the intuitions of the discrete-time algorithms in [26, 27] are not well understood. Recently, [31, 37] have proposed an ordinary differential equation with $O(\frac{1}{t^2})$ convergence, and have given a better understanding of design intuitions of the accelerated optimization methods. [2] has further shown that the best convergence rate for continuous-time accelerated algorithm is $o(\frac{1}{t^2})$ for well-tuned parameters. To deal with nonsmooth functions, [32] has generalized the results in [2] to differential inclusions by replacing gradients with subdifferentials. [30] further proposed alternative ordinary differential equations for Nesterov's accelerated methods, which are called high-resolution ODE models because they use Hessians of cost functions to distinguish between Nesterov's accelerated gradient method for strongly convex functions and Polyak's heavy-ball method. To extend the accelerated flow to primal-dual cases, [17, 50] proposed a primal-dual flow with an $O(\frac{1}{t^2})$ convergence rate for optimization problems subject to affine constraints, and applied the flow to network optimization. [49] has further extended the algorithm in [50] to saddle point problems. However, distributed accelerated continuous-time algorithms for two-subnetwork zero-sum games with nonsmooth and non-strongly convex cost functions are rarely reported.

The main shortcoming of existing works for distributed two-subnetwork zero-sum games is that they only have $O(1/t)$ or asymptotic convergence rates under convex/concave cost functions. This shortcoming motivates this paper, where we aim to propose an accelerated first-order method with a convergence rate $O(1/t^2)$ for distributed two-subnetwork zero-sum games without the strong convexity or uniform convexity assumption. The main contributions of this paper can be stated as follows:

1. This paper proposes a distributed accelerated continuous-time algorithm for a class of two-subnetwork nonsmooth zero-sum games with bilinear coupling structures, by combining Nesterov's momentum technique and the saddle point dynamics. It is shown that the proposed algorithm has an optimal convergence rate of order $O(1/t^2)$ (in terms of the duality gap function) by choosing suitable parameters. To our best knowledge, this is the fastest convergence rate on distributed algorithms for two subnetwork nonsmooth zero-sum games without strongly-convex and strongly-concave assumption on cost functions.

2. This paper shows that the proposed algorithm is well-defined and valid in the presence of nonsmooth cost functions, using the Lyapunov stability theory of differential inclusions. In contrast, most accelerated continuous-time algorithms focus on differentiable cases. The proposed algorithm extends the recent works in [49, 50] to two-subnetwork zero-sum games, and relaxes the assumption on the differentiability of cost functions in these works.

The paper is organized as follows. Section 2 gives mathematical preliminaries. Section 3 presents the two-subnetwork nonsmooth zero-sum game with bilinear couplings, and proposes a distributed accelerated algorithm for seeking a Nash equilibrium. Section 4 proves that the proposed algorithm owns an $O(1/t^2)$ rate of convergence, which is faster than existing results. Finally, Section 5 gives concluding remarks.

2. MATHEMATICAL PRELIMINARY

2.1. Notation

$\mathbb{R}$ denotes the set of real numbers; $\overline{\mathbb{R}}_+$ denotes the set of nonnegative real numbers; $\mathbb{R}^n$ denotes the set of n-dimensional real column vectors; $\mathbb{R}^{n\times m}$ denotes the set of n-by-m real matrices; I_n denotes the $n \times n$ identity matrix; $(\cdot)^{\mathrm{T}}$ denotes transpose. We write $\operatorname{rank} A$ for the rank of the matrix A, $\operatorname{range}(A)$ for the range of the matrix A, $\ker(A)$ for the kernel of the matrix A, $\mathbf{1}_n$ for the $n \times 1$ ones vector, $\mathbf{0}_n$ for the $n \times 1$ zeros vector, and $A \otimes B$ for the Kronecker product of matrices A and B. Furthermore, $\|\cdot\|$ denotes the Euclidean norm; $\|\cdot\|_p$ denotes the p-norm where $p \geq 1$; $A > 0$ ($A \geq 0$) denotes that matrix $A \in \mathbb{R}^{n\times n}$ is positive definite (positive semi-definite); $\operatorname{dist}(p, \mathcal{M})$ denotes the distance from a point p to the set $\mathcal{M}$, that is, $\operatorname{dist}(p, \mathcal{M}) \triangleq \inf_{x\in\mathcal{M}} \|p - x\|$; $x(t) \to \mathcal{M}$ as $t \to \infty$ denotes that $x(t)$ approaches the set $\mathcal{M}$, that is, for each $\epsilon > 0$ there exists $T > 0$ such that $\operatorname{dist}(x(t), \mathcal{M}) < \epsilon$ for all $t > T$.

Let $f : \Omega \to \mathbb{R}$ be a continuous function. Function f is said to be (strictly) convex if $f(\lambda s_2 + (1 - \lambda)s_1) \leq (<) \lambda f(s_2) + (1 - \lambda)f(s_1)$ for any $s_1, s_2 \in \Omega$ and $\lambda \in (0, 1)$. If f is convex, $\partial f(x)$ denotes the subdifferential of $f(\cdot)$. For convenience, we define

$\partial[-f(x)] = -\partial f(x)$ when f is a convex function without causing confusions. Consider an absolutely continuous function $x(\cdot) \in \Omega$. It is shown in [3] that a convex function f satisfies $\frac{\mathrm{d}}{\mathrm{d}t} f(x(t)) = v^\top \dot{x}(t)$ for all $v \in \partial f(x(t))$ and almost all t. A vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$ is *Lipschitz continuous* if there exists $\kappa > 0$ such that $\|f(s_2) - f(s_1)\| \leqslant \kappa \|s_2 - s_1\|$ for all $s_1, s_2 \in \mathbb{R}^n$. Let $f : \overline{\mathbb{R}}_+ \to \overline{\mathbb{R}}_+$. $f(t) = O(1/t^n)$ denotes that there exist constant $C > 0$ and $t_0 \in \overline{\mathbb{R}}_+$ such that $f(t) \leq Ct^{-n}$ for all $t \geq t_0$.

2.2. Graph Theory

An undirected graph $\mathcal{G}$ is denoted by $\mathcal{G}(\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V} = \{1, \ldots, n\}$ is a set of nodes, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is a set of edges, and $A = [a_{i,j}] \in \mathbb{R}^{n \times n}$ is the *adjacency matrix* such that $a_{i,j} = a_{j,i} > 0$ if $(j, i) \in \mathcal{E}$ and $a_{i,j} = 0$ otherwise. The *Laplacian matrix* is $L_n = D - A$, where $D \in \mathbb{R}^{n \times n}$ is diagonal with $D_{i,i} = \sum_{j=1}^{n} a_{i,j}$, $i \in \{1, \ldots, n\}$. If the graph $\mathcal{G}$ is undirected and connected, then $L_n = L_n^\top \geq 0$, $\operatorname{rank} L_n = n - 1$, and $\ker(L_n) = \{k\mathbf{1}_n : k \in \mathbb{R}\}$.

2.3. Nonsmooth Analysis

Consider a second-order differential inclusion given by

$$\ddot{u}(t) + \partial F(u(t)) \ni h(t, u(t), \dot{u}(t)), \tag{1}$$

where $t \geq t_0 \geq 0$, $u(t_0) = u_0 \in \mathbb{R}^d$, $\dot{u}(t_0) = \dot{u}_0 \in \mathbb{R}^d$, $h : \overline{\mathbb{R}}_+ \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$, $F : \mathbb{R}^d \to \mathbb{R}$, and $u \in \mathbb{R}^d$ is the state variable.

Definition 2.1. (Paoli [28]) A function $u : [t_0, T] \to \mathbb{R}^d$ is a solution of (1) with $u(t_0) = u_0$ and $\dot{u}(t_0) = \dot{u}_0$ if

1. u is Lipschitz continuous,
2. $\dot{u}$ is an absolutely continuous function,
3. function (1) holds almost everywhere in $[t_0, T]$.

To guarantee the existence of solutions to system (1), the following assumption is needed.

Assumption 2.2.

(1) Function h is a continuous function from $[t_0, T] \times \mathbb{R}^d \times \mathbb{R}^d$ to $\mathbb{R}^d$ and is Lipschitz continuous in its last two arguments uniformly with respect to the first one.

(2) Function F is a convex function from $\mathbb{R}^d$ to $\mathbb{R} \cup \{+\infty\}$, lower bounded, non-identically equal to $+\infty$ and lower semicontinuous.

Then, we provide a result (a special case of Theorem 3.1 of [28]) on the existence of solutions to system (1).

Lemma 2.3. Let Assumption 2.2 hold. For any initial condition $(u_0, \dot{u}_0)$, system (1) has a solution in the sense of Definition 2.2.

3. PROBLEM FORMULATION AND DISTRIBUTED ALGORITHM

In this section, we present a class of two-subnetwork zero-sum games with nonsmooth cost functions and bilinear couplings, and propose a distributed algorithm with damping and increasing coefficients.

3.1. Problem Description

Consider two subnetworks $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1, A_1)$ and $\mathcal{G}_2(\mathcal{V}_2, \mathcal{E}_2, A_2)$ with $\mathcal{V}_1 = \{1, \ldots, n_1\}$ and $\mathcal{V}_2 = \{1, \ldots, n_2\}$. Define strategy variables, feasibility sets, and payoff functions of the nonsmooth subnetwork bilinear zero-sum game as follows.

- **Strategic variables** of networks $\mathcal{G}_1$ and $\mathcal{G}_2$ are $\mathbf{x} = [x_1^\top, \ldots, x_{n_1}^\top]^\top \in \mathbb{R}^{pn_1}$ and $\mathbf{y} = [y_1^\top, \ldots, y_{n_2}^\top]^\top \in \mathbb{R}^{qn_2}$, respectively. The local variable of agent i in $\mathcal{G}_1$ ($\mathcal{G}_2$) is $x_i \in \mathbb{R}^p$ ($y_i \in \mathbb{R}^q$).

- **Feasible sets** for strategic variables of $\mathcal{G}_1$ and $\mathcal{G}_2$ are

$$\Omega_1 = \{\mathbf{x} \in \mathbb{R}^{pn_1} : x_i = x_j, \forall i, j \in \mathcal{V}_1\}, \quad \Omega_2 = \{\mathbf{y} \in \mathbb{R}^{qn_2} : y_i = y_j, \forall i, j \in \mathcal{V}_2\}.$$

  That is, the strategy variables of agents in the same subnetwork are required to be consensus.

- The **payoff functions** of networks $\mathcal{G}_1$ ($\mathcal{G}_2$) is $U(\cdot)$ ($-U(\cdot)$), where function $U : \mathbb{R}^{pn_1} \times \mathbb{R}^{qn_2} \to \mathbb{R}$ is given by

$$U(\mathbf{x}, \mathbf{y}) = \tilde{f}(\mathbf{x}) + \mathbf{y}^\top H \mathbf{x} - \tilde{g}(\mathbf{y}), \tag{2}$$

  with $\tilde{f}(\mathbf{x}) = \sum_{i=1}^{n_1} f_i(x_i)$, $\tilde{g}(\mathbf{y}) = \sum_{i=1}^{n_2} g_i(y_i)$, $\mathbf{y}^\top H \mathbf{x} = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} y_j^\top H_{i,j} x_i$. Matrix $H_{i,j} \in \mathbb{R}^{q \times p}$ is a non-zero matrix if agent i of $\mathcal{G}_1$ and agent j of $\mathcal{G}_2$ can observe each other's strategy variable. Define $\mathcal{N}_0$ as the set of connected edges between two subnetworks such that $(i, j) \in \mathcal{N}_0$ if $H_{i,j} \neq 0_{q \times p}$. Agent $i \in \{1, \ldots, n_1\}$ of $\mathcal{G}_1$ knows the information of $f_i(\cdot)$, x_i, $H_{i,j}$, and y_j for $(i, j) \in \mathcal{N}_0$; agent $j \in \{1, \ldots, n_2\}$ of $\mathcal{G}_2$ knows the information of $g_j(\cdot)$, $H_{i,j}$, y_j, and x_i for $(i, j) \in \mathcal{N}_0$.

In this setup, subnetwork $\mathcal{G}_1$ chooses its strategy $\mathbf{x} \in \Omega_1$ to minimize $U(\mathbf{x}, \mathbf{y})$, and subnetwork $\mathcal{G}_2$ chooses $\mathbf{y} \in \Omega_2$ to maximize $U(\mathbf{x}, \mathbf{y})$. Then we give the definition of a Nash equilibrium of the game.

Definition 3.1. A strategy $(\mathbf{x}^*, \mathbf{y}^*) \in \Omega_1 \times \Omega_2$ is a *Nash equilibrium* of the two-subnetwork zero-sum game if

$$\mathbf{x}^* \in \arg\min_{\mathbf{x} \in \Omega_1} U(\mathbf{x}, \mathbf{y}^*), \quad \mathbf{y}^* \in \arg\max_{\mathbf{y} \in \Omega_2} U(\mathbf{x}^*, \mathbf{y}).$$

The two-subnetwork zero-sum game with nonsmooth cost functions and bilinear couplings is reformulated as

$$\min_{\mathbf{x} \in \mathbb{R}^{pn_1}} \max_{\mathbf{y} \in \mathbb{R}^{qn_2}} U(\mathbf{x}, \mathbf{y}) \tag{3a}$$

$$\text{s.t.}\, \mathbf{L}_1 \mathbf{x} = 0_{pn_1}, \quad \mathbf{L}_2 \mathbf{y} = 0_{qn_2}, \tag{3b}$$

where $\mathbf{L}_1 = L_1 \otimes I_p$, $\mathbf{L}_2 = L_2 \otimes I_q$, and L_1 and L_2 are Laplacian matrices of $\mathcal{G}_1$ and $\mathcal{G}_2$.

The objective of this paper is to propose a distributed algorithm for subnetworks $\mathcal{G}_1$ and $\mathcal{G}_2$ to solve (3), in a distributed manner that each agent $i \in \mathcal{V}_1$ of $\mathcal{G}_1$ ($j \in \mathcal{V}_2$ of $\mathcal{G}_2$) only communicates with its neighbors in the same subnetwork and observes strategy of agents j, $(i,j) \in \mathcal{N}_0$ (agent i, $(i,j) \in \mathcal{N}_0$) in the other network.

Remark 3.2. Problem (3) models distributed optimization problems with parameter uncertainties and adversaries. It becomes the widely studied distributed optimization problem [8, 9, 22] if variable $\mathbf{y}$ is removed. Practical applications include the distributed optimal consensus of multiple vehicles [33, 39] and distributed control of redundant mobile manipulators [38]. It also arises in distributed adversarial resource allocation of multiple communication channels [15, 24], in which one sub-network allocates signal power and one adversarial sub-network gives noises.

Remark 3.3. In problem (3), we do not consider set constraints for easy analysis. Suppose there are set constraints $x_i \in \mathcal{X}_i$ and $y_i \in \mathcal{Y}_i$, where $\mathcal{X}_i$ and $\mathcal{Y}_i$ are convex sets (whose projection operators are computationally simple). One can use exact penalty techniques in [5, Proposition 1.5.3] to deal with the set constraints as in [51].

The following assumption is standard for distributed algorithms.

Assumption 3.4.

(1) Graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are connected and undirected.

(2) Functions $f_i(\cdot)$ and $g_j(\cdot)$ are convex, continuous, and lower bounded for all $i \in \{1, \ldots, n_1\}$ and $j \in \{1, \ldots, n_2\}$.

(3) Problem (3) has at least one Nash equilibrium point.

Next, we establish the sufficient and necessary conditions, which are a special case of [48, Theorem 4.1], for a Nash equilibrium of problem (3).

Lemma 3.5. Under Assumption 3.4, a point $(\mathbf{x}^*, \mathbf{y}^*) \in \Omega_1 \times \Omega_2 \subset \mathbb{R}^{pn_1} \times \mathbb{R}^{qn_2}$ is a Nash equilibrium of problem (3) if and only if there exist $\lambda^* \in \mathbb{R}^{pn_1}$ and $\mu^* \in \mathbb{R}^{qn_2}$ such that

$$\mathbf{L}_1 \mathbf{x}^* = 0_{pn_1}, \tag{4a}$$

$$\mathbf{L}_2 \mathbf{y}^* = 0_{qn_2}, \tag{4b}$$

$$\partial \tilde{f}(\mathbf{x}^*) + H^\top \mathbf{y}^* + \mathbf{L}_1 \lambda^* \ni 0_{pn_1}, \tag{4c}$$

$$\partial \tilde{g}(\mathbf{y}^*) - H\mathbf{x}^* + \mathbf{L}_2 \mu^* \ni 0_{qn_2}. \tag{4d}$$

Define a Lagrangian function as

$$S(\mathbf{x}, \mu, \mathbf{y}, \lambda) = U(\mathbf{x}, \mathbf{y}) + \lambda^\top \mathbf{L}_1 \mathbf{x} - \mu^\top \mathbf{L}_2 \mathbf{y} + \frac{1}{2}\mathbf{x}^\top \mathbf{L}_1 \mathbf{x} - \frac{1}{2}\mathbf{y}^\top \mathbf{L}_2 \mathbf{y}. \tag{5}$$

If Assumption 3.4 holds, S is convex with respect to $(\mathbf{x}, \mu)$ and concave with respect to $(\mathbf{y}, \lambda)$, and hence, a point $(\mathbf{x}^*, \mu^*, \mathbf{y}^*, \lambda^*) \in \mathbb{R}^{pn_1} \times \mathbb{R}^{qn_2} \times \mathbb{R}^{qn_2} \times \mathbb{R}^{pn_1}$ satisfies (4) if and only if

$$S(\mathbf{x}, \mu, \mathbf{y}^*, \lambda^*) \geq S(\mathbf{x}^*, \mu^*, \mathbf{y}^*, \lambda^*) \geq S(\mathbf{x}^*, \mu^*, \mathbf{y}, \lambda). \tag{6}$$

3.2. Distributed Algorithm

The algorithm for agent i of $\mathcal{G}_1$ and agent j of $\mathcal{G}_2$ is proposed as

$$\begin{aligned}
\ddot{x}_i(t) \in & -\frac{\alpha_{1,i}}{t}\dot{x}_i(t) - \partial f_i(x_i(t)) - \sum_{j=1}^{n_2} H_{i,j}^\top\big(y_j(t) + \frac{t}{2}\dot{y}_j(t)\big) \\
& - \sum_{k=1}^{n_1} a_{i,k}\big(\lambda_i(t) + \frac{t}{2}\dot{\lambda}_i(t) - \lambda_j(t) - \frac{t}{2}\dot{\lambda}_j(t)\big) - \sum_{k=1}^{n_1} a_{i,k}\big(x_i(t) - x_j(t)\big), \quad (7a)\\
\ddot{\lambda}_i(t) = & -\frac{\alpha_{1,i}}{t}\dot{\lambda}_i(t) + \sum_{k=1}^{n_1} a_{i,k}\big(x_i(t) + \frac{t}{2}\dot{x}_i(t) - x_j(t) - \frac{t}{2}\dot{x}_j(t)\big), \quad (7b)\\
\ddot{y}_j(t) \in & -\frac{\alpha_{2,j}}{t}\dot{y}_j(t) - \partial g_j(y_j(t)) + \sum_{i=1}^{n_1} H_{i,j}\big(x_i(t) + \frac{t}{2}\dot{x}_i(t)\big) \\
& - \sum_{k=1}^{n_1} a_{i,k}\big(\mu_i(t) + \frac{t}{2}\dot{\mu}_i(t) - \mu_j(t) - \frac{t}{2}\dot{\mu}_j(t)\big) - \sum_{k=1}^{n_2} a_{j,k}\big(y_j(t) - y_k(t)\big), \quad (7c)\\
\ddot{\mu}_j(t) = & -\frac{\alpha_{2,j}}{t}\dot{\mu}_j(t) + \sum_{k=1}^{n_2} a_{j,k}\big(y_j(t) + \frac{t}{2}\dot{y}_j(t) - y_k(t) - \frac{t}{2}\dot{y}_k(t)\big), \quad (7d)
\end{aligned}$$

where $t \geq t_0 > 0$, $i \in \{1, \dots, n_1\}$, $j \in \{1, \dots, n_2\}$, $\alpha_{1,i} > 3$, $\alpha_{2,j} > 3$, $x_i(0) = x_{i,0} \in \mathbb{R}^p$, $\dot{x}_i(0) = \dot{x}_{i,0} \in \mathbb{R}^p$, $y_j(0) = y_{j,0} \in \mathbb{R}^q$, $\dot{y}_j(0) = \dot{y}_{j,0} \in \mathbb{R}^q$, $\lambda_i(0) = \lambda_{i,0} \in \mathbb{R}^p$, $\dot{\lambda}_i(0) = \dot{\lambda}_{i,0} \in \mathbb{R}^p$, $\mu_j(0) = \mu_{j,0} \in \mathbb{R}^q$, $\dot{\mu}_j(0) = \dot{\mu}_{j,0} \in \mathbb{R}^q$, and $a_{i,k}^1$ ($a_{j,k}^2$) is the (i,k)th ((j,k)th) element of the adjacency matrix of graph $\mathcal{G}_1$ ($\mathcal{G}_2$).

Remark 3.6. In algorithm (7), $\alpha_{1,i} > 3$ and $\alpha_{2,j} > 3$ are suitable choices, which can be proven following the techniques in [49].

Remark 3.7. The dual variable $y_j(t)$ in algorithm (7) arises from the specific property of the game and makes the convergence challenging. Unlike distributed algorithms with second-order dynamics in [14, 36, 47], algorithm (7) incorporates derivatives multiplied by time-varying gains to obtain a convergence rate of $O(\frac{1}{t^2})$, which are proven to be faster than existing ones with the convergence rate of $O(\frac{1}{t})$ in Section 4.

For convenience, we omit time t in the remaining of this paper without confusion. Let $D_1 = \text{diag}[\alpha_{1,1}, \dots, \alpha_{1,n_1}] \in \mathbb{R}^{n_1 \times n_1}$, $D_2 = \text{diag}[\alpha_{2,1}, \dots, \alpha_{2,n_2}] \in \mathbb{R}^{n_2 \times n_2}$, and define $\mathbf{D}_1 = D_1 \otimes I_p$, $\mathbf{D}_2 = D_2 \otimes I_q$. Algorithm (7) has an equivalent form

$$\begin{aligned}
\ddot{\mathbf{x}} &\in -\frac{1}{t}\mathbf{D}_1\dot{\mathbf{x}} - \partial\tilde{f}(\mathbf{x}) - \mathbf{L}_1\mathbf{x} - H^\top\big(\mathbf{y} + \frac{t}{2}\dot{\mathbf{y}}\big) - \mathbf{L}_1(\lambda + \frac{t}{2}\dot{\lambda}), \quad (8a)\\
\ddot{\lambda} &= -\frac{1}{t}\mathbf{D}_1\dot{\lambda} + \mathbf{L}_1\big(\mathbf{x} + \frac{t}{2}\dot{\mathbf{x}}\big), \quad (8b)\\
\ddot{\mathbf{y}} &\in -\frac{1}{t}\mathbf{D}_2\dot{\mathbf{y}} - \partial\tilde{g}(\mathbf{y}) - \mathbf{L}_2\mathbf{y} + H(\mathbf{x} + \frac{t}{2}\dot{\mathbf{x}}) - \mathbf{L}_2\big(\mu + \frac{t}{2}\dot{\mu}\big), \quad (8c)\\
\ddot{\mu} &= -\frac{1}{t}\mathbf{D}_2\dot{\mu} + \mathbf{L}_2\big(\mathbf{y} + \frac{t}{2}\dot{\mathbf{y}}\big). \quad (8d)
\end{aligned}$$

4. MAIN RESULT

This section presents the convergence results of algorithm (8).

4.1. Existence of Solutions

If $u = [\mathbf{x}^\top, \mu^\top, \mathbf{y}^\top, \lambda^\top]^\top \in \mathbb{R}^{2pn_1+2qn_2}$, $F(u) = \tilde{f}(\mathbf{x}) + \tilde{g}(\mathbf{y})$, and

$$h(t,u,\dot{u}) = \begin{bmatrix} -\frac{1}{t}\mathbf{D}_1\dot{\mathbf{x}} - \mathbf{L}_1\mathbf{x} - H^\top\big(\mathbf{y} + \frac{t}{2}\dot{\mathbf{y}}\big) - \mathbf{L}_1(\lambda + \frac{t}{2}\dot{\lambda}) \\ -\frac{1}{t}\mathbf{D}_1\dot{\lambda} + \mathbf{L}_1\big(\mathbf{x} + \frac{t}{2}\dot{\mathbf{x}}\big) \\ -\frac{1}{t}\mathbf{D}_2\dot{\mathbf{y}} - \mathbf{L}_2\mathbf{y} + H(\mathbf{x} + \frac{t}{2}\dot{\mathbf{x}}) - \mathbf{L}_2\big(\mu + \frac{t}{2}\dot{\mu}\big) \\ -\frac{1}{t}\mathbf{D}_2\dot{\mu} + \mathbf{L}_2\big(\mathbf{y} + \frac{t}{2}\dot{\mathbf{y}}\big) \end{bmatrix},$$

then algorithm (8) can be rewritten as (1). For any interval $[t_0, T]$, one can verify that Assumption 2.2 holds if Assumption 3.4 is satisfied. Hence, it follows from Lemma 2.3 that algorithm (8) has a solution over any bounded interval $[t_0, T]$ under Assumption 3.4.

4.2. Convergence of Algorithm

For ease of notion, we define $\mathbf{z} \triangleq [\mathbf{x}^\top, \mu^\top, \mathbf{y}^\top, \lambda^\top]^\top \in \mathbb{R}^{2pn_1+2qn_2}$. Let $\mathbf{z}^* \in \mathbb{R}^{2pn_1+2qn_2}$ be an equilibrium of algorithm (8). Clearly, $\mathbf{z}^*$ satisfies (4), and hence, there exist $f_{\mathbf{x}^*} \in \partial\tilde{f}(\mathbf{x}^*)$ and $g_{\mathbf{y}^*} \in \partial\tilde{g}(\mathbf{y}^*)$ that

$$\mathbf{L}_1\mathbf{x}^* = 0_{pn_1}, \tag{9a}$$

$$\mathbf{L}_2\mathbf{y}^* = 0_{qn_2}, \tag{9b}$$

$$f_{\mathbf{x}^*} + H^\top\mathbf{y}^* + \mathbf{L}_1\lambda^* = 0_{pn_1}, \tag{9c}$$

$$g_{\mathbf{y}^*} - H\mathbf{x}^* + \mathbf{L}_2\mu^* = 0_{qn_2}. \tag{9d}$$

By Lemma 3.5, $(\mathbf{x}^*, \mathbf{y}^*)$ is a Nash equilibrium of problem (3). Define the duality gap function as

$$\begin{aligned} G(\mathbf{x},\mathbf{y}) &= S(\mathbf{x},\mu,\mathbf{y}^*,\lambda^*) - S(\mathbf{x}^*,\mu^*,\mathbf{y},\lambda) \\ &= \tilde{f}(\mathbf{x}) - \tilde{f}(\mathbf{x}^*) + \tilde{g}(\mathbf{y}) - \tilde{g}(\mathbf{y}^*) + \mathbf{y}^{*\top}H(\mathbf{x}-\mathbf{x}^*) - (\mathbf{y}-\mathbf{y}^*)^\top H\mathbf{x}^* \\ &\quad + \lambda^{*\top}\mathbf{L}_1\mathbf{x} + \mu^{*\top}\mathbf{L}_2\mathbf{y} + \frac{1}{2}\mathbf{x}^\top\mathbf{L}_1\mathbf{x} + \frac{1}{2}\mathbf{y}^\top\mathbf{L}_2\mathbf{y}. \end{aligned}$$

By the property of saddle points (see (4)) and the positive semi-definiteness of $\mathbf{L}_1, \mathbf{L}_2$,

$$G(\mathbf{x},\mathbf{y}) \geq \frac{1}{2}\mathbf{x}^\top\mathbf{L}_1\mathbf{x} + \frac{1}{2}\mathbf{y}^\top\mathbf{L}_2\mathbf{y} \geq 0, \quad \forall(\mathbf{x},\mathbf{y}) \in \mathbb{R}^{pn_1} \times \mathbb{R}^{qn_2}. \tag{10}$$

Define function

$$V(t,\mathbf{z},\dot{\mathbf{z}}) = V_1(t,\mathbf{x},\mathbf{y}) + V_2(t,\mathbf{x},\dot{\mathbf{x}}) + V_3(t,\mu,\dot{\mu}) + V_4(t,\mathbf{y},\dot{\mathbf{y}}) + V_5(t,\lambda,\dot{\lambda}), \tag{11}$$

where $V_1 = \frac{1}{2}t^2 G(\mathbf{x},\mathbf{y})$,

$$V_2 = \|\mathbf{x} + \frac{t}{2}\dot{\mathbf{x}} - \mathbf{x}^*\|^2 + \frac{1}{2}(\mathbf{x}-\mathbf{x}^*)^\top(\mathbf{D}_1 - 3I_{pn_1})(\mathbf{x}-\mathbf{x}^*),$$

$$V_3 = \|\mu + \frac{t}{2}\dot{\mu} - \mu^*\|^2 + \frac{1}{2}(\mu - \mu^*)^\top(\mathbf{D}_2 - 3I_{qn_2})(\mu - \mu^*),$$
$$V_4 = \|\mathbf{y} + \frac{t}{2}\dot{\mathbf{y}} - \mathbf{y}^*\|^2 + \frac{1}{2}(\mathbf{y} - \mathbf{y}^*)^\top(\mathbf{D}_2 - 3I_{qn_2})(\mathbf{y} - \mathbf{y}^*),$$

and

$$V_5 = \|\lambda + \frac{t}{2}\dot{\lambda} - \lambda^*\|^2 + \frac{1}{2}(\lambda - \lambda^*)^\top(\mathbf{D}_1 - 3I_{pn_1})(\lambda - \lambda^*).$$

The following Lemma shows that function $V(\cdot)$ along algorithm (8) is non-increasing to time.

Lemma 4.1. Suppose that Assumption 3.4 holds. The derivative of $V(\cdot)$ along algorithm (8) satisfies

$$\begin{aligned}\dot{V}(t,\mathbf{z},\dot{\mathbf{z}}) &= -\frac{1}{2}\mathbf{x}^\top\mathbf{L}_1\mathbf{x} - \frac{1}{2}\mathbf{y}^\top\mathbf{L}_2\mathbf{y} - \dot{\mathbf{x}}^\top(\mathbf{D}_1 - 3I_{pn_1})\dot{\mathbf{x}} \\ &\quad -\dot{\mathbf{y}}^\top(\mathbf{D}_2 - 3I_{qn_2})\dot{\mathbf{y}} - \dot{\lambda}^\top(\mathbf{D}_1 - 3I_{pn_1})\dot{\lambda} - \dot{\mu}^\top(\mathbf{D}_2 - 3I_{qn_2})\dot{\mu} \le 0. \quad (12)\end{aligned}$$

The proof of Lemma 4.1 is given in Appendix.

Then we give the following theorem to show the convergence rate of the proposed algorithm in terms of the gap function $G(\cdot)$.

Theorem 4.2. Suppose that Assumption 3.4 holds. Let $\mathbf{z}(t)$ be a trajectory of algorithm (8).

(i) The trajectory of $(\mathbf{z}(t), t\dot{\mathbf{z}}(t))$ is bounded for any $t \ge t_0$.

(ii) The trajectory $(\mathbf{x}(t), \mathbf{y}(t))$ satisfies that $G(\mathbf{x}(t), \mathbf{y}(t)) = O(\frac{1}{t^2})$, $\mathbf{x}^\top\mathbf{L}_1\mathbf{x} = O(\frac{1}{t^2})$, and $\mathbf{y}^\top\mathbf{L}_2\mathbf{y} = O(\frac{1}{t^2})$.

(iii) If, in addition, $f_i(\cdot)$ and $g_j(\cdot)$ are strictly convex for some $i \in \{1, \dots, n_1\}$ and $j \in \{1, \dots, n_2\}$, $(\mathbf{x}(t), \mathbf{y}(t))$ converges to the Nash equilibrium of problem (3).

Proof. (i) Consider function (11). It is clear that V is radially unbounded and positive definite with respect to $(\mathbf{z} - \mathbf{z}^*, t\dot{\mathbf{z}})$ for all $t \ge t_0$. It follows from Lemma 4.1 that $\dot{V}(t,\mathbf{z},\dot{\mathbf{z}}) \le 0$, and hence, the trajectory of $(\mathbf{z}(t), t\dot{\mathbf{z}}(t))$ is bounded for $t \ge t_0$.

(ii) Since $\dot{V}(t,\mathbf{z}(t),\dot{\mathbf{z}}(t)) \le 0$, then $V(t,\mathbf{z}(t),\dot{\mathbf{z}}(t)) \le m_0 \triangleq V(t_0,\mathbf{z}(t_0),\dot{\mathbf{z}}(t_0))$. Recall that $V_i(\cdot) \ge 0$ for $i \in \{1, \dots, 5\}$. It follows from (11) and definition of $V_1(\cdot)$ that

$$0 \le \frac{1}{2}G(\mathbf{x}(t), \mathbf{y}(t)) = \frac{1}{t^2}V_1(t, \mathbf{x}(t), \mathbf{y}(t)) \le \frac{1}{t^2}m_0.$$

Hence, $G(\mathbf{x}(t), \mathbf{y}(t)) = O(\frac{1}{t^2})$. It follows from (10) that $\mathbf{x}^\top\mathbf{L}_1\mathbf{x} = O(\frac{1}{t^2})$, and $\mathbf{y}^\top\mathbf{L}_2\mathbf{y} = O(\frac{1}{t^2})$.

(iii) Since the trajectory of $\mathbf{z}$ is bounded for $t \ge t_0$, $\mathbf{x}^\top\mathbf{L}_1\mathbf{x} = O(\frac{1}{t^2})$ and $\mathbf{y}^\top\mathbf{L}_2\mathbf{y} = O(\frac{1}{t^2})$ imply that $(\mathbf{x}(t), \mathbf{y}(t)) \to \Omega_1 \times \Omega_2$ as $t \to \infty$. If $f_i(\cdot)$ and $g_j(\cdot)$ are strictly convex for some $i \in \{1, \dots, n_1\}$ and $j \in \{1, \dots, n_2\}$, then the minimizer of $G(\cdot)$ subject to the constraint set $\Omega_1 \times \Omega_2$ is unique. Thus, $(\mathbf{x}(t), \mathbf{y}(t)) \to (\mathbf{x}^*, \mathbf{y}^*)$ as $t \to \infty$. It follows from Lemma 3.5 that $(\mathbf{x}^*, \mathbf{y}^*)$ is a Nash equilibrium of problem (3). The proof is thus completed. □

Remark 4.3. Theorem 4.2 shows that algorithm (8) has an $O(\frac{1}{t^2})$ convergence rate (in terms of the duality gap function), which is faster than existing results in [13, 15, 24], and relaxes the strongly monotone requirements in [10, 18, 46]. The main challenge for establishing the convergence rate of the proposed algorithm is finding a suitable Lyapunov candidate function. Augmented Lagrangian functionss and quadratic functions are combined to overcome this issue.

Remark 4.4. Theorem 4.2 (iii) assumes that there are some $i \in \{1, \ldots, n_1\}$ and $j \in \{1, \ldots, n_2\}$ such that $f_i(\cdot)$ and $g_j(\cdot)$ are strictly convex for showing the convergence of the proposed algorithm. One main contribution is relaxing the assumptions of uniform convexity [6] and strong convexity [43]. If the assumption is further relaxed as either $f_i(\cdot)$ or $g_j(\cdot)$ is strictly convex, we have some convergence property in a weaker sense. Without loss of generality, if $f_i(\cdot)$ is strictly convex for some $i \in \{1, \ldots, n_1\}$, then $\mathbf{x}(t) \to \mathbf{x}^*$ and for every unbounded positive increasing sequence $\{t_k\}_{k=1}^{\infty}$ such that $\hat{\mathbf{y}} = \lim_{t_k \to \infty} \mathbf{y}(t_k)$, $(\mathbf{x}^*, \hat{\mathbf{y}})$ is a Nash equilibrium of problem (3).

4.3. Comparison with Existing Results

In this subsection, we compare the rate of convergence of algorithm (8) with that of an algorithm proposed in [15]. Specifically, the design in [15] for this problem is

$$\begin{aligned}
\dot{x}_i(t) \in &- \partial f_i(x_i(t)) - \sum_{j=1}^{n_2} H_{i,j}^\top y_j(t) \\
&- \sum_{k=1}^{n_1} a_{i,k}\big(\lambda_i(t) - \lambda_j(t)\big) - \sum_{k=1}^{n_1} a_{i,k}\big(x_i(t) - x_j(t)\big), \qquad (13a)\\
\dot{\lambda}_i(t) =& \sum_{k=1}^{n_1} a_{i,k}\big(x_i(t) - x_j(t)\big), \qquad (13b)\\
\dot{y}_j(t) \in &- \partial g_j(y_j(t)) + \sum_{i=1}^{n_1} H_{i,j} x_i(t) \\
&- \sum_{k=1}^{n_1} a_{i,k}\big(\mu_i(t) - \mu_j(t)\big) - \sum_{k=1}^{n_2} a_{j,k}\big(y_j(t) - y_k(t)\big), \qquad (13c)\\
\dot{\mu}_j(t) =& \sum_{k=1}^{n_2} a_{j,k}\big(y_j(t) - y_k(t)\big). \qquad (13d)
\end{aligned}$$

The convergence and boundedness of algorithm (13) are proved in [15]. Define the ergodic trajectory as $\hat{\mathbf{x}}(t) = \frac{1}{t}\int_0^t \mathbf{x}(s)\,\mathrm{d}s$, $\hat{\mathbf{y}}(t) = \frac{1}{t}\int_0^t \mathbf{y}(s)\,\mathrm{d}s$, $\hat{\lambda}(t) = \frac{1}{t}\int_0^t \lambda(s)\,\mathrm{d}s$, and $\hat{\mu}(t) = \frac{1}{t}\int_0^t \mu(s)\,\mathrm{d}s$. We further show the rate of convergence of algorithm (13) for problem (3), which is not obtained in [15].

Lemma 4.5. Suppose that Assumption 3.4 holds. Let $(\mathbf{x}(t), \mathbf{y}(t), \lambda(t), \mu(t))$ be a trajectory of algorithm (13). The the ergodic trajectory $(\hat{\mathbf{x}}(t), \hat{\mathbf{y}}(t), \hat{\lambda}(t), \hat{\mu}(t))$ satisfies $G(\hat{\mathbf{x}}(t), \hat{\mathbf{y}}(t)) = O(\frac{1}{t})$.

P r o o f . Define function $V(\mathbf{x},\mathbf{y},\lambda,\mu)=\frac{1}{2}\|\mathbf{x}-\mathbf{x}^*\|^2+\frac{1}{2}\|\mathbf{y}-\mathbf{y}^*\|^2+\frac{1}{2}\|\lambda-\lambda^*\|^2+\frac{1}{2}\|\mu-\mu^*\|^2$, where $(\mathbf{x}^*,\mathbf{y}^*,\lambda^*,\mu^*)$ is an equilibrium of algorithm (13), equivalently, $(\mathbf{x}^*,\mathbf{y}^*)$ is a Nash equilibrium of problem (3). The derivative of V is

$$\dot{V} \;=\; (\mathbf{x}-\mathbf{x}^*)^\top\dot{\mathbf{x}}+(\mathbf{y}-\mathbf{y}^*)^\top\dot{\mathbf{y}}+(\lambda-\lambda^*)^\top\dot{\lambda}+(\mu-\mu^*)^\top\dot{\mu}.$$

Note that S is convex (concave) with respect $\mathbf{x}$ and μ ($\mathbf{y}$ and λ). It follows from algorithm (13) that

$$\begin{aligned}(\mathbf{x}-\mathbf{x}^*)^\top\dot{\mathbf{x}}+(\mu-\mu^*)^\top\dot{\mu} &\le S\big(\mathbf{x}^*,\mu^*,\mathbf{y},\lambda\big)-S\big(\mathbf{x},\mu,\mathbf{y},\lambda\big),\\ (\mathbf{y}-\mathbf{y}^*)^\top\dot{\mathbf{y}}+(\lambda-\lambda^*)^\top\dot{\lambda} &\le S\big(\mathbf{x},\mu,\mathbf{y},\lambda\big)-S\big(\mathbf{x},\mu,\mathbf{y}^*,\lambda^*\big).\end{aligned}$$

Hence, $\dot{V}(\mathbf{x},\mathbf{y},\lambda,\mu)\le S\big(\mathbf{x}^*,\mu^*,\mathbf{y},\lambda\big)-S\big(\mathbf{x},\mu,\mathbf{y}^*,\lambda^*\big)=-G(\mathbf{x},\mathbf{y})\le 0$. Hence

$$V(\mathbf{x}(t),\mathbf{y}(t),\lambda(t),\mu(t))-V(\mathbf{x}_0,\mathbf{y}_0,\lambda_0,\mu_0)\le-\int_0^t G(\mathbf{x}(s),\mathbf{y}(s))\,\mathrm{d}s\le 0.$$

Since $V(\cdot)\ge 0$, $\int_0^t G(\mathbf{x}(s),\mathbf{y}(s))\,\mathrm{d}s\le V(\mathbf{x}_0,\mathbf{y}_0,\lambda_0,\mu_0)$. It follows from Jensen's inequality and the convex-concave property of S that

$$\begin{aligned}S\big(\mathbf{x}^*,\mu^*,\hat{\mathbf{y}}(t),\hat{\lambda}(t)\big)&\ge\frac{1}{t}\int_0^t S\big(\mathbf{x}^*,\mu^*,\mathbf{y}(s),\lambda(s)\big)\,\mathrm{d}s,\\ S\big(\hat{\mathbf{x}}(t),\hat{\mu}(t),\mathbf{y}^*,\lambda^*\big)&\le\frac{1}{t}\int_0^t S\big(\mathbf{x}(s),\mu(s),\mathbf{y}^*,\lambda^*\big)\,\mathrm{d}s.\end{aligned}$$

By definition of $G(\cdot)$, we have

$$G(\hat{\mathbf{x}}(t),\hat{\mathbf{y}}(t))\le\frac{1}{t}\int_0^t G(\mathbf{x}(s),\mathbf{y}(s))\,\mathrm{d}s.$$

Hence, $G(\hat{\mathbf{x}}(t),\hat{\mathbf{y}}(t))\le\frac{1}{t}V(\mathbf{x}_0,\mathbf{y}_0,\lambda_0,\mu_0)$. ◊ □

Remark 4.6. By comparing Theorem 4.2 and Lemma 4.5, we show that the proposed method (8) has a faster convergence rate compared with existing results in [15].

4.4. Numerical Simulation

Consider the two-subnetwork zero-sum game (3). We take $p=q=2$, $n_1=n_2=25$, and $H=I_{50}$.

Case 1 (Convex functions): $f_i(x_i)=\frac{2}{3}\log(1+(x_{i,1}+i)^2)+\frac{1}{3}\log(1+(x_{i,2}-i)^2)$, $g_i(y_i)=\log((y_{i,1}-iy_{i,2})^2+1/2)$ for $i\in\{1,\ldots,25\}$.

Case 2 (Quadratic convex functions): $f_i(x_i)=(x_{i,1}-x_{i,2}+i)^2$ and $g_i(y_i)=(y_{i,1}+y_{i,2}+i)^2$ for $i\in\{1,\ldots,25\}$.

We approximate the numerical trajectories of proposed algorithms using function ODE45 of Matlab. For case 1, Figure 1 gives trajectories of $S(\cdot)$ (defined in (5)) along algorithm (8) with $\alpha = 2, 4$, and shows that $\alpha = 4$ gives a better performance. Figure 2 compares algorithm (8) and a primal-dual algorithm in [15] for case 1. Figure 2 indicates that the proposed algorithm has a better performance than the primal-dual method in [15] for case 1. Figure 3 shows that, for case 2 whose cost functions are quadratic, the primal-dual method in [15] has a better convergence performance than the proposed algorithm (8). The reason for this is that the primal-dual method in [15] is an affine algorithm whose convergence rate is linear.

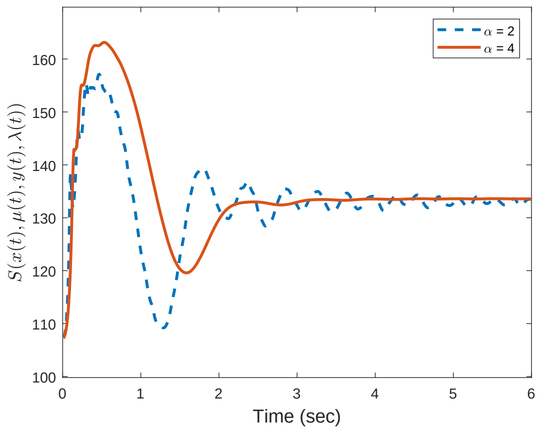

Fig. 1. Trajectories of $S(\cdot)$ along algorithm (8) with $\alpha = 2, 4$ for case 1.

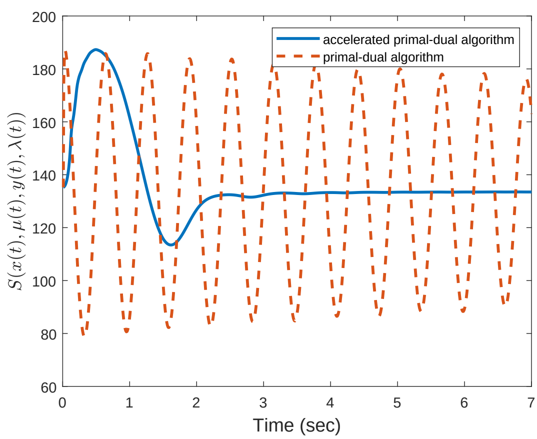

Fig. 2. Trajectories of $S(\cdot)$ along algorithm (8) and primal-dual algorithm in [15] for case 1.

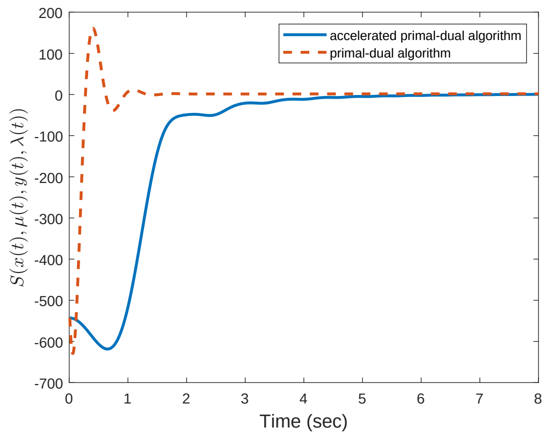

Fig. 3. Trajectories of $S(\cdot)$ along algorithm (8) and primal-dual algorithm in [15] for case 2.

5. CONCLUSION

This paper has proposed a distributed accelerated first-order algorithm owning $O(1/t^2)$ convergence to a Nash equilibrium in a class of two-subnetwork zero-sum games. This paper has established a rate of convergence $O(1/t^2)$ by choosing suitable parameters, using the stability theory of differential inclusions. The proposed distributed algorithm is considerably different in algorithm design and convergence analysis. The algorithm has been proven to converge faster than existing results under mild conditions. Future research includes incorporating complex constraints and developing discrete-time counterparts of the proposed continuous-time algorithm.

APPENDIX. PROOF OF LEMMA 4.1

It follows from algorithm (8) that there exist $f_{\mathbf{x}} \in \partial \tilde{f}(\mathrm{x})$ and $g_{\mathbf{y}} \in \partial \tilde{g}(\mathrm{y})$ such that

$$\ddot{\mathbf{x}} = -\frac{1}{t}\mathbf{D}_1\dot{\mathbf{x}} - f_{\mathbf{x}} - H^\top(\mathbf{y} + \frac{t}{2}\dot{\mathbf{y}}) - \mathbf{L}_1\big(\lambda + \frac{t}{2}\dot{\lambda}\big) - \mathbf{L}_1\mathbf{x}, \tag{14a}$$

$$\ddot{\mathbf{y}} = -\frac{1}{t}\mathbf{D}_2\dot{\mathbf{y}} - g_{\mathbf{y}} + H(\mathbf{x} + \frac{t}{2}\dot{\mathbf{x}}) - \mathbf{L}_2\big(\mu + \frac{t}{2}\dot{\mu}\big) - \mathbf{L}_2\mathbf{y}. \tag{14b}$$

The derivative of V at time t is

$$\dot{V}(t,\mathbf{z},\dot{\mathbf{z}}) = \dot{V}_1(t,\mathbf{x},\mathbf{y}) + \dot{V}_2(t,\mathbf{x},\dot{\mathbf{x}}) + \dot{V}_3(t,\mu,\dot{\mu}) + \dot{V}_4(t,\mathbf{y},\dot{\mathbf{y}}) + \dot{V}_5(t,\lambda,\dot{\lambda}), \tag{15}$$

where

$$\begin{aligned}
\dot{V}_1 &= t[S(\mathbf{x},\mu,\mathbf{y}^*,\lambda^*)-S(\mathbf{x}^*,\mu^*,\mathbf{y},\lambda)]+\frac{1}{2}t^2[f_{\mathbf{x}}+H^\top\mathbf{y}^*+\mathbf{L}_1\lambda^*+\mathbf{L}_1\mathbf{x}]^\top\dot{\mathbf{x}} \\
&\quad +\frac{1}{2}t^2[g_{\mathbf{y}}-H\mathbf{x}^*+\mathbf{L}_2\mu^*+L_2\mathbf{y}]^\top\dot{\mathbf{y}}, \qquad (16)\\
\dot{V}_2 &= 2(\mathbf{x}+\frac{1}{2}t\dot{\mathbf{x}}-\mathbf{x}^*)^\top(3\dot{\mathbf{x}}+t\ddot{\mathbf{x}})+2(\alpha-3)(\mathbf{x}-\mathbf{x}^*)^\top\dot{\mathbf{x}}, \qquad (17)\\
\dot{V}_3 &= 2(\mu+\frac{1}{2}t\dot{\mu}-\mu^*)^\top(3\dot{\mu}+t\ddot{\mu})+2(\alpha-3)(\mu-\mu^*)^\top\dot{\mu}, \qquad (18)\\
\dot{V}_4 &= 2(\mathbf{y}+\frac{1}{2}t\dot{\mathbf{y}}-\mathbf{y}^*)^\top(3\dot{\mathbf{y}}+t\ddot{\mathbf{y}})+2(\alpha-3)(\mathbf{y}-\mathbf{y}^*)^\top\dot{\mathbf{y}}, \qquad (19)\\
\dot{V}_5 &= 2(\lambda+\frac{1}{2}t\dot{\lambda}-\lambda^*)^\top(3\dot{\lambda}+t\ddot{\lambda})+2(\alpha-3)(\lambda-\lambda^*)^\top\dot{\lambda}. \qquad (20)
\end{aligned}$$

Plugging (9c) and (9d) in (16) gives

$$\begin{aligned}
\dot{V}_1 &= t[S(\mathbf{x},\mu,\mathbf{y}^*,\lambda^*)-S(\mathbf{x}^*,\mu^*,\mathbf{y},\lambda)]+\frac{1}{2}t^2[f_{\mathbf{x}}-f_{\mathbf{x}^*}+\mathbf{L}_1\mathbf{x}]^\top\dot{\mathbf{x}} \\
&\quad +\frac{1}{2}t^2[g_{\mathbf{y}}-g_{\mathbf{y}^*}+\mathbf{L}_2\mathbf{y}]^\top\dot{\mathbf{y}}. \qquad (21)
\end{aligned}$$

It follows from (9), (14), and mathematical derivations that

$$\begin{aligned}
\dot{V}_2 &= (\mathbf{x}-\mathbf{x}^*)^\top(\mathbf{D}_1\dot{\mathbf{x}}+t\ddot{\mathbf{x}})+0.5t^2\dot{\mathbf{x}}^\top(3\dot{\mathbf{x}}+\ddot{\mathbf{x}}) \\
&= -t(\mathbf{x}-\mathbf{x}^*)^\top\Big(f_{\mathbf{x}}+H^\top\mathbf{y}+\mathbf{L}_1\lambda+\mathbf{L}_1\mathbf{x}\Big)-\frac{t^2}{2}(\mathbf{x}-\mathbf{x}^*)^\top(H^\top\dot{\mathbf{y}}+\mathbf{L}_1\dot{\lambda}) \\
&\quad -0.5t\dot{\mathbf{x}}^\top(\mathbf{D}_1-3I_{pn_1})\dot{\mathbf{x}}-\frac{1}{2}t^2\dot{\mathbf{x}}^\top\Big(f_{\mathbf{x}}+H^\top\mathbf{y}+\mathbf{L}_1\lambda+\mathbf{L}_1\mathbf{x}\Big) \\
&\quad -0.25t^3\dot{\mathbf{x}}^\top(H^\top\dot{\mathbf{y}}+\mathbf{L}_1\dot{\lambda}) \\
&= -t(\mathbf{x}-\mathbf{x}^*)^\top(f_{\mathbf{x}}-f_{\mathbf{x}^*})-t(\mathbf{x}-\mathbf{x}^*)^\top H^\top(\mathbf{y}-\mathbf{y}^*)-t\mathbf{x}^\top\mathbf{L}_1(\lambda-\lambda^*) \\
&\quad -t\mathbf{x}^\top\mathbf{L}_1\mathbf{x}-0.5t^2(\mathbf{x}-\mathbf{x}^*)^\top(H^\top\dot{\mathbf{y}}+\mathbf{L}_1\dot{\lambda})-0.5t\dot{\mathbf{x}}^\top(\mathbf{D}_1-3I_{pn_1})\dot{\mathbf{x}} \\
&\quad -0.25t^3\dot{\mathbf{x}}^\top(H^\top\dot{\mathbf{y}}+\mathbf{L}_1\dot{\lambda})-0.5t^2\dot{\mathbf{x}}^\top\mathbf{L}_1\mathbf{x}-0.5t^2\dot{\mathbf{x}}^\top(f_{\mathbf{x}}-f_{\mathbf{x}^*}) \\
&\quad -0.5t^2\dot{\mathbf{x}}^\top H^\top(\mathbf{y}-\mathbf{y}^*)-0.5t^2\dot{\mathbf{x}}^\top\mathbf{L}_1(\lambda-\lambda^*), \qquad (22)
\end{aligned}$$

$$\begin{aligned}
\dot{V}_3 &= t(\mu-\mu^*)^\top\mathbf{L}_2\mathbf{y}+\frac{1}{2}t^2(\mu-\mu^*)^\top\mathbf{L}_2\dot{\mathbf{y}}+0.5t^2\dot{\mu}^\top\mathbf{L}_2\mathbf{y} \\
&\quad +0.5t\dot{\mu}^\top(3I_{qn_2}-\mathbf{D}_2)\dot{\mu}+0.25t^3\dot{\mu}^\top\mathbf{L}_2\dot{\mathbf{y}}, \qquad (23)
\end{aligned}$$

$$\begin{aligned}\dot{V}_4 &= (\mathbf{y}-\mathbf{y}^*)^\top\Big(-tg_{\mathbf{y}}+tH(\mathbf{x}-a)-t\mathbf{L}_2\mu-t\mathbf{L}_2\mathbf{y}\Big)\\&\quad+\frac{t^2}{2}(\mathbf{y}-\mathbf{y}^*)^\top(H\dot{\mathbf{x}}-\mathbf{L}_2\dot{\mu})-0.5t\dot{\mathbf{y}}^\top(\mathbf{D}_2-3I_{qn_2})\dot{\mathbf{y}}\\&\quad+0.25t^3\dot{\mathbf{y}}^\top(H\dot{\mathbf{x}}-\mathbf{L}_2\dot{\mu})+0.5t^2\dot{\mathbf{y}}^\top\Big(-g_{\mathbf{y}}+H(\mathbf{x}-a)-\mathbf{L}_2\mu-\mathbf{L}_2\mathbf{y}\Big)\\&= -t(\mathbf{y}-\mathbf{y}^*)^\top(g_{\mathbf{y}}-g_{\mathbf{y}^*})+t(\mathbf{y}-\mathbf{y}^*)^\top H(\mathbf{x}-\mathbf{x}^*)-t\mathbf{y}^\top\mathbf{L}_2(\mu-\mu^*)\\&\quad-t\mathbf{y}^\top\mathbf{L}_2\mathbf{y}+0.5t^2(\mathbf{y}-\mathbf{y}^*)^\top(H\dot{\mathbf{x}}-\mathbf{L}_2\dot{\mu})-0.5t\dot{\mathbf{y}}^\top(\mathbf{D}_2-3I)\dot{\mathbf{y}}\\&\quad+0.25t^3\dot{\mathbf{y}}^\top(H\dot{\mathbf{x}}-\mathbf{L}_2\dot{\mu})-0.5t^2\dot{\mathbf{y}}^\top(g_{\mathbf{y}}-g_{\mathbf{y}^*})+0.5t^2\dot{\mathbf{y}}^\top H(\mathbf{x}-\mathbf{x}^*)\\&\quad-0.5t^2\dot{\mathbf{y}}^\top\mathbf{L}_2(\mu-\mu^*)-0.5t^2\dot{\mathbf{y}}^\top\mathbf{L}_2\mathbf{y},\end{aligned}\tag{24}$$

$$\begin{aligned}\dot{V}_5 &= t(\lambda-\lambda^*)^\top\mathbf{L}_1x+\frac{1}{2}t^2(\lambda-\lambda^*)^\top\mathbf{L}_1\dot{\mathbf{x}}+0.25t^3\dot{\lambda}^\top\mathbf{L}_1\dot{\mathbf{x}}\\&\quad-0.5t\dot{\lambda}^\top(\mathbf{D}_1-3I_{pn_1})\dot{\lambda}+0.5t^2\dot{\lambda}^\top\mathbf{L}_1\mathbf{x}.\end{aligned}\tag{25}$$

By summing up (21) – (25) and simplifying items, we have

$$\begin{aligned}\dot{V} &= t[S(\mathbf{x},\mu,\mathbf{y}^*,\lambda^*)-S(\mathbf{x}^*,\mu^*,\mathbf{y},\lambda)]-t(\mathbf{x}-\mathbf{x}^*)^\top(f_{\mathbf{x}}-f_{\mathbf{x}^*})\\&\quad-t\mathbf{x}^\top\mathbf{L}_1\mathbf{x}-t(\mathbf{y}-\mathbf{y}^*)^\top(g_{\mathbf{y}}-g_{\mathbf{y}^*})-t\mathbf{y}^\top\mathbf{L}_2\mathbf{y}-0.5t\dot{\mathbf{x}}^\top(\mathbf{D}_1-3I_{pn_1})\dot{\mathbf{x}}\\&\quad-0.5t\dot{\mathbf{y}}^\top(\mathbf{D}_2-3I_{qn_2})\dot{\mathbf{y}}-0.5t\dot{\lambda}^\top(\mathbf{D}_1-3I_{pn_1})\dot{\lambda}-0.5t\dot{\mu}^\top(\mathbf{D}_2-3I_{qn_2})\dot{\mu}\\&= t\mathbf{A}-0.5t\mathbf{B},\end{aligned}\tag{26}$$

where

$$\begin{aligned}\mathbf{A} &= \tilde{f}(\mathbf{x})-\tilde{f}(\mathbf{x}^*)+\tilde{g}(\mathbf{y})-\tilde{g}(\mathbf{y}^*)+\mathbf{y}^{*\top}H\mathbf{x}-\mathbf{y}^\top H\mathbf{x}^*+\lambda^{*\top}\mathbf{L}_1\mathbf{x}+\mu^{*\top}\mathbf{L}_2\mathbf{y}\\&\quad-\frac{1}{2}\mathbf{x}^\top\mathbf{L}_1\mathbf{x}-\frac{1}{2}\mathbf{y}^\top\mathbf{L}_2\mathbf{y}-(\mathbf{x}-\mathbf{x}^*)^\top(f_{\mathbf{x}}-f_{\mathbf{x}^*})-(\mathbf{y}-\mathbf{y}^*)^\top(g_{\mathbf{y}}-g_{\mathbf{y}^*}),\\\mathbf{B} &= \dot{\mathbf{x}}^\top(\mathbf{D}_1-3I_{pn_1})\dot{\mathbf{x}}+\dot{\mathbf{y}}^\top(\mathbf{D}_2-3I_{qn_2})\dot{\mathbf{y}}+\dot{\lambda}^\top(\mathbf{D}_1-3I_{pn_1})\dot{\lambda}\mu\\&\quad+\dot{\mu}^\top(\mathbf{D}_2-3I_{qn_2})\dot{\geq}0.\end{aligned}\tag{27}$$

By (9c) and (9d), $\mathbf{A}$ can be rewritten as

$$\begin{aligned}\mathbf{A} &= \tilde{f}(\mathbf{x})-\tilde{f}(\mathbf{x}^*)-(\mathbf{x}-\mathbf{x}^*)^\top f_{\mathbf{x}}-\frac{1}{2}\mathbf{x}^\top\mathbf{L}_1\mathbf{x}+\tilde{g}(\mathbf{y})-\tilde{g}(\mathbf{y}^*)\\&\quad-(\mathbf{y}-\mathbf{y}^*)^\top g_{\mathbf{y}}-\frac{1}{2}\mathbf{y}^\top\mathbf{L}_2\mathbf{y},\end{aligned}$$

where $f_{\mathbf{x}}\in\partial\tilde{f}(\mathrm{x})$ and $g_{\mathbf{y}}\in\partial\tilde{g}(\mathrm{y})$. Because $\tilde{f}$ and $\tilde{g}$ are convex, it is clear that $\tilde{f}(\mathbf{x})-\tilde{f}(\mathbf{x}^*)-(\mathbf{x}-\mathbf{x}^*)^\top f_{\mathbf{x}}\leq 0$ and $\tilde{g}(\mathbf{y})-\tilde{g}(\mathbf{y}^*)-(\mathbf{y}-\mathbf{y}^*)^\top g_{\mathbf{y}}\leq 0$. Note that $\mathbf{L}_1\geq 0$ and $\mathbf{L}_2\geq 0$. It follows that

$$\mathbf{A}\leq-\frac{1}{2}\mathbf{x}^\top\mathbf{L}_1\mathbf{x}-\frac{1}{2}\mathbf{y}^\top\mathbf{L}_2\mathbf{y}\leq 0.\tag{28}$$

Then, combining (26) – (28) proves the conclusion of the lemma.

ACKNOWLEDGEMENT

This work was partially supported by the Talent Program of Guangdong Province (2021QN02Z107) and the National Natural Science Foundation of China under Grant 61991413.

(Received October 23, 2022)

REFERENCES

[1] M. S. Alkousa, A. V. Gasnikov, D. M. Dvinskikh, D. A. Kovalev, and F. S. Stonyakin: Accelerated methods for saddle-point problem. Comput. Math. and Math. Phys. *60* (2020), 1787–1787. DOI:10.1134/S0965542520110020

[2] H. Attouch, Z. Chbani, and H. Riahi: Rate of convergence of the Nesterov accelerated gradient method in the subcritical case $\alpha \leq 3$. ESAIM: COCV *25* (2019), 2. DOI:10.1051/cocv/2017083

[3] J. P. Aubin and A. Cellina: Differential Inclusions. Springer-Verlag, Berlin 1984.

[4] A. Beck and M. Teboulle: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. *21* (2009), 1, 183–202. DOI:10.1137/080716542

[5] D. P. Bertsekas: Convex Optimization Algorithms. Athena Scientific, Belmont 2015.

[6] A. Chambolle and T. Pock: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. *40* (2011), 1, 120–145. DOI:10.1007/s10851-010-0251-1

[7] Y. Chen, G. Lan, and Y. Ouyang: Optimal primal-dual methods for a class of saddle point problems. SIAM J. Control Optim. *24* (2014), 4, 1779–1814. DOI:10.1137/130919362

[8] S. Chen and S. Liang: Distributed optimization for multi-agent system over unbalanced graphs with linear convergence rate. Kybernetika *56* (2020), 3, 559–577. DOI:10.31857/S0044452920070438

[9] Z. Chen and S. Liang: Distributed aggregative optimization with quantized communication. Kybernetika *58* (2022), 1, 123–144. DOI:10.14736/kyb-2022-1-0123

[10] G. Chen, Y. Ming, Y. Hong, and P. Yi: Distributed algorithm for ε-generalized Nash equilibria with uncertain coupled constraints. Automatica *123* (2021), 109313. DOI:10.1016/j.automatica.2020.109313

[11] J. Chen, J. Sun, and G. Wang: From unmanned systems to autonomous intelligent systems. Engineering *8* (2022), 1–5. DOI:10.1155/2022/1426724

[12] A. Cherukuri, B. Gharesifard, and J. Cortés: Saddle-point dynamics: Conditions for asymptotic stability of saddle points. SIAM J. Control Optim. *55* (2017), 1, 486–511. DOI:10.1137/15M1026924

[13] Z. Deng: Distributed generalized Nash equilibrium seeking algorithm for nonsmooth aggregative games. Automatica *132* (2021), 109794. DOI:10.1016/j.automatica.2021.109794

[14] Z. Deng: Distributed Nash equilibrium seeking for aggregative games with second-order nonlinear players. Automatica *135* (2022), 109980. DOI:10.1016/j.automatica.2021.109980

[15] B. Gharesifard and J. Cortés: Distributed convergence to Nash equilibria in two-network zero-sum games. Automatica *49* (2013), 6, 1683–1692. DOI:10.1016/j.automatica.2013.02.062

[16] R. Goebel: Stability and robustness for saddle-point dynamics through monotone mappings. Syst. Control. Lett. *108* (2017), 16–22. DOI:10.1016/j.sysconle.2017.07.014

[17] X. He, R. Hu, and Y. P. Fang: Convergence rates of inertial primal-dual dynamical methods for separable convex optimization problems. SIAM J. Control Optim. *59* (2021), 5, 3278–3301. DOI:10.1137/20M1355379

[18] S. Huang, J. Lei, and Y. Hong: A linearly convergent distributed nash equilibrium seeking algorithm for aggregative games. IEEE Trans. Automat. Control , 2022. DOI:10.1109/TAC.2022.3154356

[19] S. Huang, J. Lei, Y. Hong, and U. V. Shanbhag: No-regret distributed learning in two-network zero-sum games. In: 60th IEEE Conference on Decision and Control (CDC), 2021, pp. 924–929.

[20] T. Ibaraki and N. Katoh: Resource Allocation Problems: Algorithmic Approaches. MIT Press, Cambridge 1988.

[21] S. Khan, M. Tufail, M. T. Khan, Z. A. Khan, J. Iqbal, and A. Wasim: A novel framework for multiple ground target detection, recognition and inspection in precision agriculture applications using a UAV. Unmanned Syst. *10* (2022), 1, 45–56.

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

[23] P. Li, J. Hu, L. Qiu, Y. Zhao, and B.K. Ghosh: A distributed economic dispatch strategy for power-water networks. IEEE Trans. Control. Netw. Syst*9* (2022), 1, 356–366. DOI:10.1109/TCNS.2021.3104103

[24] Y. Lou, Y. Hong, L. Xie, G. Shi, and K. H. Johansson: Nash equilibrium computation in subnetwork zero-sum games with switching communications. IEEE Trans. Automat. Control *61* (2016), 10, 2920–2935. DOI:10.1109/TAC.2015.2504962

[25] L. Mo, Y. Yu, G. Ren, and X. Yuan: Constrained consensus of continuous-time heterogeneous multi-agent networks with nonconvex constraints and delays. J. Syst. Sci. Complex *35* (2022), 105–122. DOI:10.1007/s11424-021-0092-6

[26] Y. Nesterov: A method of solving a convex programming problem with convergence rate $O(1/k^2)$. Soviet Math. Doklady *27* (1983), 372–376.

[27] Y. Nesterov: Smooth minimization of non-smooth functions. Math Program. *103* (2005), 1, 127–152. DOI:10.1007/s10107-004-0552-5

[28] L. A. Paoli: An existence result for vibrations with unilateral constraints: Case of a nonsmooth set of constraints. Math. Models Methods Appl. Sci. *10* (2000), 6, 815–831. DOI:10.1142/S0218202500000422

[29] Z. Peng, R. Luo, J. Hu, K. Shi, and B. K. Ghosh: Distributed optimal tracking control of discrete-time multiagent systems via event-triggered reinforcement learning. IEEE Trans. Circuits Syst. I Regular Papers *69* (2022), 9, 3689–3700.

[30] B. Shi, S. S. Du, M. I. Jordan, and W. J. Su: Understanding the acceleration phenomenon via high-resolution differential equations. Math. Program. *195* (2022), 79–148. DOI:10.1007/s10107-021-01681-8

[31] W. Su, S. Boyd, and E. J. Candes: A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. Adv. Neural Inf. Process. Syst. *3* (2015), 1, 2510–2518.

[32] A. Vassilis, A. Jean-Francois, and D. Charles: The differential inclusion modeling FISTA algorithm and optimality of convergence $b \leq 3$. SIAM J. Control. Optim. *29* (2018), 1, 551–574. DOI:10.1137/17m1128642

[33] Q. Wang, J. Chen, B. Xin, and X. Zeng: Distributed optimal consensus for Euler-Lagrange systems based on event-triggered control. IEEE Trans. Syst. Man Cybern. Syst. *51* (2021), 7, 4588–4598. DOI:10.1109/TSMC.2019.2944857

[34] M. Wang, L. Li, Q. Dai, and F. Shi: Resource allocation based on DEA and non-cooperative game. J. Syst. Sci. Complex *34* (2022), 2231–2249. DOI:10.1007/s11424-021-0259-1

[35] Y. Wang, P. Lin, and H. Qin: Distributed classification learning based on nonlinear vector support machines for switching networks. Kybernetika *53* (2017), 4, 595–611. DOI:10.14736/kyb-2017-4-0595

[36] D. Wang, Z. Wang, Z. Wu, and W. Wang: Distributed convex optimization for nonlinear multi-agent systems disturbed by a second-order stationary process over a digraph. Sci. China Inf. Sci. *65* (2022), 132201.

[37] A. Wibisono, A. C. Wilson, and M. I. Jordan: A variational perspective on accelerated methods in optimization. Proc. Natl. Acad. Sci. *113* (2016), 47, 7351–7358. DOI:10.13109/kont.2016.47.2.113

[38] C. Wu, H. Fang, Q. Yang, X. Zeng, Y. Wei, and J. Chen: Distributed cooperative control of redundant mobile manipulators with safety constraints. IEEE Trans. Cybernet. *53* (2023), 2, 1195–1207. DOI:10.1109/TCYB.2021.3104044

[39] Y. Wu, Q. Liang, Y. Zhao, J. Hu, and L. Xiang: Distributed estimation based output consensus control of heterogeneous leader-follower systems with antagonistic interactions. Sci. China Inf. Sci. *66* (2023), 139204.

[40] Y. Wu, Q. Liang, Y. Zhao, J. Hu, and L. Xiang: Effective distributed algorithm for solving linear matrix equations. Sci. China Inf. Sci. *66* (2023), 189202.

[41] S. Xie, L. Wang, M. H. Nazari, G. Yin, and G. Li: Distributed optimization with markovian switching targets and stochastic observation noises with applications to dc micro-grids. Sci. China Inf. Sci. *65* (2022), 222205.

[42] Y. Xu: Accelerated first-order primal-dual proximal methods for linearly constrained composite convex programming. SIAM J. Optim. *27* (2018), 3, 1459–1484. DOI:10.1137/16M1082305

[43] Y. Xu and S. Zhang: Accelerated primal-dual proximal block coordinate updating methods for constrained convex optimization. Comput. Optim. Appl. *70* (2018), 91–128. DOI:10.1007/s10589-017-9972-z

[44] R. Yang, L. Liu, and G. Feng: An overview of recent advances in distributed coordination of multi-agent systems. Unmanned Syst. 2022. DOI:10.1142/S2301385021500199

[45] S. Yang, J. Wang, and Q. Liu: Cooperative-competitive multiagent systems for distributed minimax optimization subject to bounded constraints. IEEE Trans. Automat. Control *64* (2019), 4, 358–1372. DOI:10.1109/TAC.2018.2862471

[46] M. Ye, G. Hu, and S. Xu: An extremum seeking-based approach for Nash equilibrium seeking in N-cluster noncooperative games. Automatica *114* (2020), 108815. DOI:10.1016/j.automatica.2020.108815

[47] M. Ye, J. Yin, and L. Yin: Distributed Nash equilibrium seeking for games in second-order systems without velocity measurement. IEEE Trans. Automat. Control. DOI:10.1109/TAC.2021.3131553

[48] X. Zeng, J. Chen, S. Liang, and Y. Hong: Generalized Nash equilibrium seeking strategy for distributed nonsmooth multi-cluster game. Automatica *103* (2019), 20–26. DOI:10.1016/j.automatica.2019.01.025

[49] X. Zeng, L. Dou, and J. Chen: Accelerated first-order continuous-time algorithm for solving convex-concave bilinear saddle point problem. In: 21st IFAC World Congress, Berlin 2020.

[50] X. Zeng, J. Lei, and J. Chen: Dynamical primal-dual accelerated method with applications to network optimization. IEEE Trans. Automat. Control. DOI:10.1109/TAC.2022.3152720

[51] H. Zhou, X. Zeng, and Y. Hong: Adaptive exact penalty design for constrained distributed optimization. IEEE Trans. Automat. Control *64* (2019), 11, 4661–4667. DOI:10.1109/tac.2019.2902612

Xianlin Zeng, National Key Laboratory on Autonomous Intelligent Unmanned Systems, School of Automation, Beijing Institute of Technology, Beijing, 100081*. P. R. China.*
e-mail: xianlin.zeng@bit.edu.cn

Lihua Dou, National Key Laboratory on Autonomous Intelligent Unmanned Systems, School of Automation, Beijing Institute of Technology, Beijing, 100081*. P. R. China.*
e-mail: doulihua@bit.edu.cn

Jinqiang Cui, Peng Cheng Laboratory, Shenzhen, 518055*. P. R. China.*
e-mail: cuijq@pcl.ac.cn