Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Accelerated First-Order Continuous-Time Algorithm for Solving Convex-Concave Bilinear Saddle Point Problem

Xianlin Zeng [*], **Lihua Dou** [*] and **Jie Chen** [*,**]

[*] *Key Laboratory of Biomimetic Robots and Systems,*
Beijing Advanced Innovation Center for Intelligent Robots and
Systems, Beijing Institute of Technology,
Ministry of Education, Beijing, China.
[**] *Department of Control Science and Engineering, Tongji University,*
the Shanghai Research Institute for Intelligent Autonomous Systems,
Shanghai, China.

Abstract: First-order methods have simple structures and are of great importance to big data problems because first-order methods are easy to implement in a distributed or parallel way. However, in the worst cases, first-order methods often converge at a rate $O(1/t)$, which is slow. This paper considers a class of convex-concave bilinear saddle point problems and proposes an accelerated first-order continuous-time algorithm. We design the accelerated algorithm by using both increasing and decreasing damping coefficients in the saddle point dynamics. If parameters of the proposed algorithm are proper, the algorithm owns $O(1/t^2)$ convergence without any strict or strong convexity requirement. Finally, we apply the algorithm to numerical examples to show the superior performance of the proposed algorithm over existing ones.

Keywords: Accelerated method, first-order algorithm, continuous-time algorithm, saddle-point problem.

1. INTRODUCTION

Convex-concave bilinear saddle point problems are an important model in optimization (see Ruszczynski (2006)). Many constrained optimization problems in applied mathematics and engineering, such as signal/image processing (see Beck and Teboulle (2009)), machine learning (see Boyd et al. (2011)), and distributed optimization (see Chen and Kai (2018); Kia et al. (2015); Wang and Elia (2010); Yi et al. (2015); Zeng et al. (2017)), can be cast as convex-concave bilinear saddle point problems. Due to the wide application of convex-concave bilinear saddle point problems, the design of efficient algorithms for such problems is of great importance.

Motivated by big data and distributed computation problems, first-order methods have received tremendous attention for a wide class of constrained optimization problems due to the fact that first-order algorithms only require gradient information and may drastically simplify the computation of the optimization problem. In particular, first-order methods have been applied to distributed optimization problems such as optimal consensus problems (see Kia et al. (2015); Yi et al. (2015)), resource allocation (see Yi et al. (2016)), and extended monotropic optimization problems (see Zeng et al. (2018)).

Rates of convergence are of great importance for optimization algorithms. It is well known that when the cost functions are convex, the rate of convergence of first-order primal-dual optimization algorithms is $O(1/t)$ under the worst choice of cost functions. The Nesterov accelerated method has been developed in Nesterov (1983) by using a vanishing damping coefficient to have $O(1/t^2)$ convergence rate, which was proved to be optimal in some sense (see Nemirovskii and Yudin (1983)). However, most existing accelerated results (see Attouch et al. (2015); Siegel (2019); Su et al. (2015); Wibisono et al. (2016)) focus on primal-based methods and can not be applied to the saddle point problems. In recent years, Beck and Teboulle (2009) has developed accelerated discrete-time primal-dual methods, and Xu (2017) has proposed accelerated linearized augmented Lagrangian method and an accelerated alternating direction method of multipliers for solving structured linearly constrained convex programming. However, results in Beck and Teboulle (2009) and Xu (2017) are based on the assumption that cost functions are simple with easy minimization operations or proximal mappings.

Recently, continuous-time optimization algorithm have been revisited via the Lyapunov approach for both accelerated algorithms (see Attouch et al. (2015); Siegel (2019); Su et al. (2015); Wibisono et al. (2016)) and distributed algorithms (see Kia et al. (2015); Wang and

[⋆] This work was supported by NSFC (Grant no. 61873033), Beijing Institute of Technology Research Fund Program for Young Scholars, the New Generation Artificial Intelligence Megaproject (Grant no. 2018AAA0101600), Projects of Major International (Regional) Joint Research Program NSFC (Grant no. 61720106011), and Consulting Research Project of the Chinese Academy of Engineering (Grant no. 2019-XZ-7). Email: xianlin.zeng@bit.edu.cn; doulihua@bit.edu.cn; chenjie@bit.edu.cn

Elia (2010); Yi et al. (2015); Zeng et al. (2017); Zhou et al. (2019)). On one hand, ordinary differential equations often exhibit similar convergence properties to their discrete-time counterparts without tuning the step sizes, which is typically a hard task. On the other hand, continuous-time algorithms may serve as a tool for algorithm design and analysis with explainable intuitions and ideas (see Bhaya and Kaszkurewicz (2000)).

Focusing on convex-concave bilinear saddle point problems, this paper aims to answer two questions:

(1) Can we propose an accelerated first-order continuous-time algorithm that has a faster convergence rate than $O(1/t)$?
(2) If we have an accelerated first-order continuous-time algorithm, what is the best choice for parameters to obtain an optimal convergence rate?

The contributions of this paper can be summarized as follows:

- This paper extends the Nesterov accelerated method Nesterov (1983) to the primal-dual method for convex-concave bilinear saddle point problems. To our best knowledge, this is the first accelerated first-order continuous-time method for convex-concave bilinear saddle point problems.
- By using the Lyapunov approach, we give rigorous proofs that the proposed algorithm can converge at an optimal rate of $O(1/t^2)$ by choosing proper parameters. We have shown that $\alpha > 3$ is a better choice for the parameter in the proposed algorithm and the critical value for the parameter is $\alpha = 3$. This result is consistent to the existing results in primal-based accelerated methods (see Attouch et al. (2015); Su et al. (2015); Wibisono et al. (2016)).

The paper is organized as follows. Section 2 gives the problem formulation and proposes an accelerated primal-dual continuous-time algorithm. Section 3 proves the convergence of the proposed algorithm and shows the $O(1/t^2)$ rate of convergence under some conditions. Then Section 4 shows numerical examples to verify the efficacy of the proposed algorithm. Finally, Section 5 gives concluding remarks.

2. PROBLEM FORMULATION AND ALGORITHM

In this section, we review relevant notations, present the problem formulation, and design the algorithm.

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, and $(\cdot)^{\mathrm{T}}$ denotes transpose, respectively. Furthermore, $\|\cdot\|$ denotes the Euclidean norm, $\log(\cdot)$ denotes the natural logarithm function, $\mathcal{B}_\epsilon(x), x \in \mathbb{R}^n, \epsilon > 0$, is the open ball *centered* at x with *radius* ϵ, $\mathrm{dist}(p, \mathcal{M})$ is the distance from a point p to the set $\mathcal{M}$, (that is, $\mathrm{dist}(p, \mathcal{M}) \triangleq \inf_{x\in\mathcal{M}} \|p - x\|$), $x(t) \to \mathcal{M}$ 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 $\mathrm{dist}(x(t), \mathcal{M}) < \epsilon$ for all $t > T$). Let $f : \overline{\mathbb{R}}_+ \to \overline{\mathbb{R}}_+$ be a continuous function. $f(t) = O(1/t^n)$ denotes that there exists a constant $C > 0$ such that $f(t) \le Ct^{-n}$ for all $t \ge 0$.

2.2 Problem Formulation

Consider a convex-concave bilinear saddle point problem given by

$$\min_{x\in\mathbb{R}^q} \max_{y\in\mathbb{R}^m} L(x, y), \tag{1}$$

where

$$L(x, y) \triangleq f(x) + y^\top (Ax - b) - g(y), \tag{2}$$

$A \in \mathbb{R}^{m\times q}$, $b \in \mathbb{R}^m$, and $f : \mathbb{R}^q \to \mathbb{R}$ and $g : \mathbb{R}^m \to \mathbb{R}$ are convex and twice differentiable functions. A point $(x^*, y^*) \in \mathbb{R}^q \times \mathbb{R}^m$ is said to be a *saddle point* of $L(\cdot, \cdot)$ if $L(x^*, y) \le L(x^*, y^*) \le L(x, y^*)$.

Remark 1. Convex-concave bilinear saddle point problems are a general model that has many applications. For example, one special case of convex-concave bilinear saddle point problems is the Lagrangian function of convex optimization problems with affine constraints. Applications of convex-concave bilinear saddle point problems include imaging and signal processing (see Beck and Teboulle (2009); Chambolle and Pock (2011)) and machine learning (see Boyd et al. (2011)). ◊

Throughout this paper, we have the following assumption.

Assumption 1. Problem (1) has at least one saddle point.

One existing algorithm for problem (1) is the first-order method given by the saddle point dynamics

$$\dot{x}(t) = -\nabla_{x(t)} L(x(t), y(t)), \quad x(0) = x_0, \tag{3a}$$
$$\dot{y}(t) = \nabla_{y(t)} L(x(t), y(t)), \quad y(0) = y_0. \tag{3b}$$

It is well-known that the rate of convergence for the algorithm is $O(1/t)$. In the remaining of this paper, we will propose a continuous-time algorithm with an improved rate of convergence $O(1/t^2)$.

2.3 Accelerated Algorithm

We propose an accelerated first-order continuous-time method for problem (1):

$$\ddot{x}(t) = -\frac{\alpha}{t}\dot{x}(t) - \nabla_{x(t)} L(x(t), y(t) + \frac{1}{2}t\dot{y}(t)), \tag{4a}$$
$$\ddot{y}(t) = -\frac{\alpha}{t}\dot{y}(t) + \nabla_{y(t)} L(x(t) + \frac{1}{2}t\dot{x}(t), y(t)), \tag{4b}$$

where $t \ge t_0 > 0$, $\alpha > 3$, $x(t_0) = x_0$, $\dot{x}(t_0) = \dot{x}_0$, $y(t_0) = y_0$, and $\dot{y}(t_0) = \dot{y}_0$.

For convenience, we omit time t in remaining of this paper without causing confusions. The specific form of algorithm (4) is

$$\ddot{x} = -\frac{\alpha}{t}\dot{x} - \nabla f(x) - A^\top(y + \frac{1}{2}t\dot{y}) \tag{5a}$$
$$\ddot{y} = -\frac{\alpha}{t}\dot{y} - \nabla g(y) + A(x + \frac{1}{2}t\dot{x}) - b. \tag{5b}$$

Note that $f(\cdot)$ and $g(\cdot)$ are twice differentiable. Functions $\nabla f(\cdot)$ and $\nabla g(\cdot)$ are locally Lipschitz continuous. It follows from (Haddad and Chellaboina, 2008, Theorem 2.38, pp. 96) that algorithm (5) has a unique trajectory. The initial time $t_0 > 0$ avoids the singularity of the damping coefficient $\frac{\alpha}{t}$ at zero. Although algorithm (5) uses $t\dot{x}(t)$ and $t\dot{y}(t)$ in the righthand side, Section 3 will show that $t\dot{x}(t)$ and $t\dot{y}(t)$ are bounded as $t \to \infty$. Hence, $\ddot{x}(t)$ and $\ddot{y}(t)$ are bounded and algorithm (5) is well defined.

Remark 2. In this design, the use of derivative information $\frac{1}{2}t\dot{x}(t)$ and $\frac{1}{2}t\dot{y}(t)$ is not intuitive. However, it is important in our design and is shown to be effective for convergence in the analysis as demonstrated later. From the control perspective, it may be viewed as a "derivative feedback" design and plays a role as damping terms. From the optimization perspective, $\frac{1}{2}t\dot{x}(t)$ and $\frac{1}{2}t\dot{y}(t)$ point to the future moving direction of $x(t)$ and $y(t)$. Thus, algorithm (5) uses the "estimated future" position $x(t) + \frac{1}{2}t\dot{x}(t)$ and $y(t) + \frac{1}{2}t\dot{y}(t)$ in this design. ◊

3. MAIN RESULTS

In this section, we prove convergence properties of the proposed method and show that $\alpha > 3$ is the optimal choice for the proposed algorithm.

3.1 *Analysis of Algorithm* (4)

In this subsection, we give rigorous analysis of the convergence properties for proposed algorithm (4).

Let $(x^*, y^*) \in \mathbb{R}^q \times \mathbb{R}^m$. It follows from the Karush-Kuhn-Tucker (KKT) optimality condition (Theorem 3.34 of Ruszczynski (2006)) that $(x^*, y^*) \in \mathbb{R}^q \times \mathbb{R}^m$ is a solution to problem (1) (a saddle point to function $L(x, y)$) if and only if

$$0_q = \nabla f(x^*) + A^\top y^*, \tag{6a}$$

$$0_m = -\nabla g(y^*) + Ax^* - b. \tag{6b}$$

Then the following theorem shows the well posedness and the convergence rate of algorithm (4).

Theorem 1. Suppose Assumption 1 holds. Let $(x(t), y(t))$ be a trajectory of algorithm (4).

(i) The trajectory of $(x(t), y(t), t\dot{x}(t), t\dot{y}(t))$ is bounded for $t \ge t_0$.

(ii) The trajectory $(x(t), y(t), \dot{x}(t), \dot{y}(t))$ satisfies the convergence properties

$$L(x(t), y^*) - L(x^*, y(t)) = O(\frac{1}{t^2}),$$

$$\|\dot{x}(t)\| = O(\frac{1}{t}), \text{ and } \|\dot{y}(t)\| = O(\frac{1}{t}).$$

(iii) For any initial condition, there exists $m_0 > 0$ such that the following integral inequalities are satisfied:

$$\int_{t_0}^{\infty} t\dot{x}^2(t) \le \frac{1}{\alpha - 3} m_0, \tag{7}$$

$$\int_{t_0}^{\infty} t\dot{y}^2(t)\mathrm{d}t \le \frac{1}{\alpha - 3} m_0. \tag{8}$$

Remark 3. In algorithm (4), α is greater than 3. By a similar line of attack as Appendix A (proof of Theorem 1), one can easily verify that $L(x(t), y^*) - L(x^*, y(t)) = O(\frac{1}{t^2})$ for $\alpha = 3$. However, if $\alpha = 3$, the trajectory of $(x(t), y(t), t\dot{x}(t), t\dot{y}(t))$ may not be bounded since $V(t, x, y, \dot{x}, \dot{y})$ defined in (A.1) is not positive definite with respect to $(x, y, t\dot{x}, t\dot{y})$ for any $t \ge t_0$. Hence, $\alpha > 3$ is a better choice than $\alpha = 3$. ◊

3.2 *Analysis of The Case* $0 < \alpha < 3$

In this subsection, we discuss the case that $0 < \alpha < 3$. Suppose $0 < \alpha < 3$. We propose a modified algorithm

$$\ddot{x}(t) = -\frac{\alpha}{t}\dot{x}(t) - \nabla_{x(t)} L(x(t), y(t) + \frac{3}{2\alpha}t\dot{y}(t)), \tag{9a}$$

$$\ddot{y}(t) = -\frac{\alpha}{t}\dot{y}(t) + \nabla_{y(t)} L(x(t) + \frac{3}{2\alpha}t\dot{x}(t), y(t)), \tag{9b}$$

where $t \ge t_0 > 0$, $0 < \alpha < 3$, $x(t_0) = x_0$, $\dot{x}(t_0) = \dot{x}_0$, $y(t_0) = y_0$, and $\dot{y}(t_0) = \dot{y}_0$. The specific form of algorithm (9) is

$$\ddot{x} = -\frac{\alpha}{t}\dot{x} - \nabla f(x) - A^\top(y + \frac{3}{2\alpha}t\dot{y}), \tag{10a}$$

$$\ddot{y} = -\frac{\alpha}{t}\dot{y} - \nabla g(y) + A(x + \frac{3}{2\alpha}t\dot{x}) - b. \tag{10b}$$

Remark 4. In algorithm (9), the gain $\frac{3}{2\alpha}$ of $t\dot{x}(t)$ and $t\dot{y}(t)$ is different from that in algorithm (4). In fact, this modification is needed for convergence analysis of algorithm (9). ◊

Then the following theorem, whose proof is given in Appendix B, shows that the rate of convergence for algorithm (9) is $O(t^{-\frac{2\alpha}{3}})$.

Theorem 2. Suppose Assumption 1 holds. Let $(x(t), y(t))$ be a trajectory of algorithm (9). Then

(i) $L(x(t), y^*) - L(x^*, y(t)) = O(\frac{1}{t^{\frac{2\alpha}{3}}})$.

(ii) $\|\dot{x}(t)\| = O(\frac{1}{t^{\frac{\alpha}{3}}})$, and $\|\dot{y}(t)\| = O(\frac{1}{t^{\frac{\alpha}{3}}})$.

Remark 5. Theorem 2 shows that algorithm (9) converges at a rate of $O(\frac{1}{t^{\frac{2\alpha}{3}}})$ for $0 < \alpha < 3$. Combining the results of Theorem 1, $\alpha > 3$ is the optimal choice. ◊

3.3 *Discussion of Results*

Results in subsections 3.1 and 3.2 show that algorithm (4) ($\alpha > 3$) is superior to algorithm (9) ($0 < \alpha < 3$) in the following aspects.

(1) The rate of convergence for algorithm (4) is $O(\frac{1}{t^2})$, which is faster than that of algorithm (9).

(2) Since the trajectory of $(x(t), y(t), t\dot{x}(t), t\dot{y}(t))$ generated by algorithm (4) is bounded, algorithm (4) is well-defined with bounded $\ddot{x}(t)$ and $\ddot{y}(t)$. However, $\ddot{x}(t)$ and $\ddot{y}(t)$ of algorithm (9) may be unbounded as $t \to \infty$.

The main challenges of proving Theorems 1 and 2 are finding appropriate Lyapunov function candidates for proposed algorithms. The design of Lyapunov function candidates in this paper is partially inspired by the results for primal-based accelerated algorithms (see Attouch et al. (2015); Su et al. (2015)). However, we have extended the design of algorithm and the analysis to primal-dual cases, which are a more general formulation. The obtained convergence rates ($O(\frac{1}{t^2})$ for the case $\alpha > 3$ and $O(\frac{1}{t^{\frac{2\alpha}{3}}})$ for the case $0 < \alpha < 3$) for convex-concave bilinear saddle point problems are consistent to that of primal-based accelerated algorithms for unconstrained convex optimization problems (see Attouch et al. (2015); Su et al. (2015)).

One should note that the accelerated performance of the proposed method does not hold for any cost function.

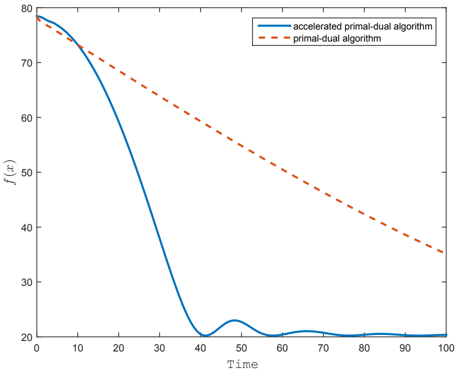

Fig. 1. Trajectories of algorithm (4) and algorithm in (11)

Suppose f and g are both strongly convex or quadratic convex functions. The primal-dual saddle point dynamics has a linear rate, which is faster than $O(\frac{1}{t^2})$ of the proposed method.

4. NUMERICAL SIMULATION

In this section, we conduct two numerical examples to show the efficacy of proposed algorithms.

Example 1. Consider an optimization problem given by $\min_{x\in\mathbb{R}^2} \hat{f}(x)$ *s.t.* $Ax = b$, where $A = [1,\ 1]$, $\hat{f}(x) = 20\log\Big[\exp([1,\ 2]x - 1)/20) + \exp([3,\ 1]x - 1)/20) + \exp([1,\ 1]x - 1)/20)\Big]$, and $b = 0$.

The augmented Lagrangian function of the optimization problem is $\hat{L}(x, y) = f(x) + y(Ax - b) + \frac{1}{2}\|Ax - b\|^2$. Recall the KKT optimality condition, one can seek a saddle point of $\hat{L}$ to solve the optimization problem using algorithm (4) by replacing L in (4) with function $\hat{L}$.

Then we compare algorithm (4) with a standard primal-dual gradient algorithm given by

$$\begin{aligned} \dot{x}(t) &= -\nabla_{x(t)}\hat{L}(x(t), y(t)), && (11a)\\ \dot{y}(t) &= \nabla_{y(t)}\hat{L}(x(t), y(t)), && (11b)\end{aligned}$$

Fig. 1 plots the simulation results and shows that the proposed algorithm has a faster convergence performance than the standard algorithm (11).

Example 2. Consider problem saddle point problem (1), where $x = [x_1, x_2]^\top \in \mathbb{R}^2$, $y = [y_1, y_2]^\top \in \mathbb{R}^2$, $f(x) = \frac{2}{3}\log(1+x_1^2) + \frac{1}{3}\log(1+x_2^2)$, $g(y) = (y_1 - y_2)^2$, $A = \begin{bmatrix} 1, & 1\\ 1, & 1\end{bmatrix}$, and $b = [0,\ 0]^\top$. The simulation results of algorithms (4) ($\alpha = 4 > 3$), algorithm (9) ($\alpha = 2 < 3$), and algorithm (3) (classic saddle point dynamics) are shown in Fig. 2. It is clear that accelerated algorithm (4) converges at a faster rate and $\alpha > 3$ is a better choice for the algorithm.

5. CONCLUSIONS

This paper has focused on designing an accelerated first-order algorithm for a class of convex-concave bilinear saddle point problems. By using increasing and decreasing damping coefficients, this paper has developed a continuous-time algorithm having $O(1/t^2)$ convergence by choosing proper parameters. For different choices of parameters, the paper has proved the correctness and convergence of the algorithm based on the Lyapunov approach.

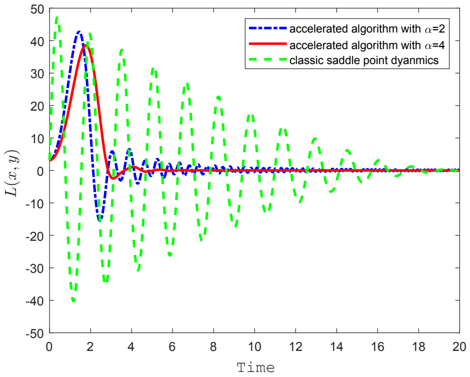

Fig. 2. Trajectories of accelerated algorithm and classic saddle point dynamics

REFERENCES

Attouch, H., Chbani, Z., and Riahi, H. (2015). Rate of convergence of the nesterov accelerated gradient method in the subcritical case $\alpha \leq 3$. *ESAIM: Control, Optimisation and Calculus of Variations*, 25.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 21(1), 183–202.

Bhaya, A. and Kaszkurewicz, E. (2000). *Control Perspectives on Numerical Algorithms and Matrix Problems.* SIAM, Philadelphia.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3, 1–122.

Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1), 120–145.

Chen, J. and Kai, S. (2018). Cooperative transportation control of multiple mobile manipulators through distributed optimization. *Sci China Inf Sci*, 61, 120201.

Haddad, W.M. and Chellaboina, V. (2008). *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach.* Princeton Univ. Press, Princeton, NJ.

Kia, S.S., Cortés, J., and Martínez, S. (2015). Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication. *Automatica*, 55, 254–264.

Nemirovskii, A. and Yudin, D. (1983). *Problem Complexity and Method Efficiency in Optimization.* John Wiley & Sons.

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

Ruszczynski, A. (2006). *Nonlinear Optimization.* Princeton University Press, Princeton, New Jersey.

Siegel, J. (2019). Accelerated first-order methods: Differential equations and Lyapunov functions. *arXiv:1903.05671v3.*

Su, W., Boyd, S., and Candes, E.J. (2015). A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. *Advances in Neural Information Processing Systems*, 3(1), 2510–2518.

Wang, J. and Elia, N. (2010). Control approach to distributed optimization. In *Proc. Allerton Conf. Commun., Control and Comput.*, 557–561. Monticello, IL.

Wibisono, A., Wilson, A.C., and Jordan, M.I. (2016). A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47), 7351–7358.

Xu, Y. (2017). Accelerated first-order primal-dual proximal methods for linearly constrained composite convex programming. *SIAM Journal on Optimization*, 27(3), 1459–1484.

Yi, P., Hong, Y., and Liu, F. (2015). Distributed gradient algorithm for constrained optimization with application to load sharing in power systems. *Systems & Control Letters*, 83, 45–52.

Yi, P., Hong, Y., and Liu, F. (2016). Initialization-free distributed algorithms for optimal resource allocation with feasibility constraints and application to economic dispatch of power systems. *Automatica*, 74, 259–269.

Zeng, X., Yi, P., and Hong, Y. (2017). Distributed continuous-time algorithm for constrained convex optimizations via nonsmooth analysis approach. *IEEE Transactions on Automatic Control*, 62(10), 5227–5233.

Zeng, X., Yi, P., Hong, Y., and Xie, L. (2018). Distributed continuous-time algorithms for nonsmooth extended monotropic optimization problems. *SIAM J. Control and Optimization*, 56(6), 3973–3993.

Zhou, H., Yu, W., Yi, P., and Hong, Y. (2019). Quantized gradient-descent algorithm for distributed resource allocation. *Unmanned Systems*, 7(2), 119–136.

Appendix A. PROOF OF THEOREM 1

Define function

$$V(t,x,y,\dot{x},\dot{y}) = V_1(t,x,y) + V_2(t,x,\dot{x}) + V_3(t,y,\dot{y}) \quad \text{(A.1)}$$

such that

$$V_1 = t^2[L(x,y^*) - L(x^*,y)], \quad \text{(A.2)}$$

$$V_2 = 2\|x + \frac{1}{2}t\dot{x} - x^*\|^2 + (\alpha-3)\|x-x^*\|^2, \quad \text{(A.3)}$$

$$V_3 = 2\|y + \frac{1}{2}t\dot{y} - y^*\|^2 + (\alpha-3)\|y-y^*\|^2, \quad \text{(A.4)}$$

where (x^*,y^*) satisfies the KKT optimality condition (6). By the property of saddle points of $L(\cdot,\cdot)$, $L(x^*,y) \le L(x,y^*)$. Hence, function V is positive definite with respect to $(x,y,t\dot{x},t\dot{y})$ for all $t \ge t_0$. Then we prove this theorem in the following steps.

Step (i): The derivatives of V_i's, $i=1,2,3$, satisfy that

$$\begin{aligned}\dot{V}_1 = {} & 2t[f(x) - f(x^*) + {y^*}^\top A(x-x^*)] \\ & +2t[g(y) - g(y^*) - (Ax^*-b)^\top(y-y^*)] \\ & +t^2[\nabla f(x) + A^\top y^*]^\top \dot{x} \\ & +t^2[\nabla g(y) - Ax^* + b]^\top \dot{y}, \end{aligned} \quad \text{(A.5)}$$

$$\begin{aligned}\dot{V}_2 = {} & 2(x + \frac{1}{2}t\dot{x} - x^*)^\top(3\dot{x} + t\ddot{x}) \\ & +2(\alpha-3)(x-x^*)^\top\dot{x}, \end{aligned} \quad \text{(A.6)}$$

$$\begin{aligned}\dot{V}_3 = {} & 2(y + \frac{1}{2}t\dot{y} - y^*)^\top(3\dot{y} + t\ddot{y}) \\ & +2(\alpha-3)(y-y^*)^\top\dot{y}. \end{aligned} \quad \text{(A.7)}$$

By plugging (4) in (A.6) and (A.7), we have

$$\begin{aligned}\dot{V}_2 = {} & -2t(x-x^*)^\top(\nabla f(x) + A^\top y) - t^2(x-x^*)^\top A^\top \dot{y} \\ & -(\alpha-3)t\|\dot{x}\|^2 - t^2\dot{x}^\top(\nabla f(x) + A^\top y) \\ & -\frac{1}{2}t^3\dot{x}^\top A^\top \dot{y}, \end{aligned} \quad \text{(A.8)}$$

$$\begin{aligned}\dot{V}_3 = {} & 2t(y-y^*)^\top(-\nabla g(y) + Ax - b) + t^2(y-y^*)^\top A\dot{x} \\ & -(\alpha-3)t\|\dot{y}\|^2 + t^2\dot{y}^\top(-\nabla g(y) + Ax - b) \\ & +\frac{1}{2}t^3\dot{x}^\top A^\top \dot{y}. \end{aligned} \quad \text{(A.9)}$$

Equations (A.8) and (A.9) can be equivalently written as

$$\begin{aligned}\dot{V}_2 = {} & -2t(x-x^*)^\top(\nabla f(x) + A^\top y^*) - t^2(x-x^*)^\top A^\top \dot{y} \\ & -2t(x-x^*)^\top A^\top(y-y^*) - (\alpha-3)t\|\dot{x}\|^2 \\ & -t^2\dot{x}^\top(\nabla f(x) + A^\top y^*) - t^2\dot{x}^\top A^\top(y-y^*) \\ & -\frac{1}{2}t^3\dot{x}^\top A^\top \dot{y}, \end{aligned} \quad \text{(A.10)}$$

$$\begin{aligned}\dot{V}_3 = {} & 2t(y-y^*)^\top(-\nabla g(y) + Ax^* - b) \\ & +2t(y-y^*)^\top A(x-x^*) + t^2(y-y^*)^\top A\dot{x} \\ & -(\alpha-3)t\|\dot{y}\|^2 + t^2\dot{y}^\top(-\nabla g(y) + Ax^* - b) \\ & +t^2\dot{y}^\top A(x-x^*) + \frac{1}{2}t^3\dot{x}^\top A^\top \dot{y}. \end{aligned} \quad \text{(A.11)}$$

By summing (A.10)-(A.11) and simplifying the terms, we have

$$\begin{aligned}\dot{V}_2 + \dot{V}_3 = {} & N + t^2\dot{y}^\top(-\nabla g(y) + Ax^* - b) \\ & -t^2\dot{x}^\top(\nabla f(x) + A^\top y^*) \\ & -2t(x-x^*)^\top(\nabla f(x) + A^\top y^*) \\ & -2t(y-y^*)^\top(\nabla g(y) - Ax^* + b), \end{aligned} \quad \text{(A.12)}$$

$$N = -(\alpha-3)t\|\dot{x}\|^2 - (\alpha-3)t\|\dot{y}\|^2. \quad \text{(A.13)}$$

It follows from (A.5) and (A.12) that

$$\begin{aligned}\dot{V} = {} & 2t[f(x) - f(x^*) - (x-x^*)^\top\nabla f(x)] \\ & 2t[g(y) - g(y^*) - (y-y^*)^\top\nabla g(y)] + N. \end{aligned} \quad \text{(A.14)}$$

Because f and g are convex, it is clear that $f(x) - f(x^*) - (x-x^*)^\top\nabla f(x) \le 0$ and $g(y) - g(y^*) - (y-y^*)^\top\nabla g(y) \le 0$. It follows from (A.13) and (A.14) that

$$\dot{V} \le -(\alpha-3)t\|\dot{x}\|^2 - (\alpha-3)t\|\dot{y}\|^2 \le 0. \quad \text{(A.15)}$$

Recall that function V is radially unbounded and positive definite with respect to $(x,y,t\dot{x},t\dot{y})$ for all $t \ge t_0$. The trajectory of $(x(t),y(t),t\dot{x}(t),t\dot{y}(t))$ is bounded for $t \ge t_0$.

Step (ii): Define

$$m_0 \triangleq V(t_0, x(t_0), y(t_0), \dot{x}(t_0), \dot{y}(t_0)). \tag{A.16}$$

Since $\dot{V} \leq 0$, then $V(t, x(t), y(t), \dot{x}(t), \dot{y}(t)) \leq m_0$. Recall that $V_1(t, x(t), y(t)) \leq V(t, x(t), y(t), \dot{x}(t), \dot{y}(t))$. It is straightforward that $L(x(t), y^*) - L(x^*, y(t)) = O(\frac{1}{t^2})$. In addition, since we have proved the boundedness of $t\dot{x}(t)$ and $t\dot{y}(t)$, it is clear that $\|\dot{x}(t)\| = O(\frac{1}{t})$, and $\|\dot{y}(t)\| = O(\frac{1}{t})$.

Step (iii): Clearly, $V(t, x(t), y(t), \dot{x}(t), \dot{y}(t)) - m_0 = \int_{t_0}^{t} \dot{V}(s, x(s), y(s), \dot{x}(s), \dot{y}(s))\mathrm{d}s$. Because $V(\cdot) \geq 0$, it follows that $-\int_{t_0}^{t} \dot{V}(s, x(s), y(s), \dot{x}(s), \dot{y}(s))\mathrm{d}s \leq m_0$. By (A.15), we have (7) and (8), where m_0 is defined in (A.16).

Appendix B. PROOF OF THEOREM 2

(i) Let $(x^*, y^*) \in \mathbb{R}^q \times \mathbb{R}^m$ satisfy the KKT optimality condition (6). Define the function

$$V(t, x, y, \dot{x}, \dot{y}) = V_1(t, x, y) + V_2(t, x, \dot{x}) + V_3(t, y, \dot{y}) \tag{B.1}$$

where $V_1 = t^{\frac{2\alpha}{3}}[L(x, y^*) - L(x^*, y)]$, $V_2 = \frac{1}{2}\left\|\frac{2\alpha}{3}t^{\frac{\alpha}{3}-1}(x - x^*) + t^{\frac{\alpha}{3}}\dot{x}\right\|^2 + \frac{(3-\alpha)\alpha}{9}t^{\frac{2\alpha}{3}-2}\|x - x^*\|^2$, and $V_3 = \frac{1}{2}\left\|\frac{2\alpha}{3}t^{\frac{\alpha}{3}-1}(y - y^*) + t^{\frac{\alpha}{3}}\dot{y}\right\|^2 + \frac{(3-\alpha)\alpha}{9}t^{\frac{2\alpha}{3}-2}\|y - y^*\|^2$.

Derivatives of V_i's, $i = 1, 2, 3$, along the trajectory of algorithm (9) satisfy that

$$\begin{aligned}\dot{V}_1 = &\frac{2\alpha}{3}t^{\frac{2\alpha}{3}-1}[f(x) - f(x^*) + {y^*}^\top A(x - x^*)]\\ &+\frac{2\alpha}{3}t^{\frac{2\alpha}{3}-1}[g(y) - g(y^*) - (Ax^* - b)^\top(y - y^*)]\\ &+t^{\frac{2\alpha}{3}}[\nabla f(x) + A^\top y^*]^\top \dot{x}\\ &+t^{\frac{2\alpha}{3}}[\nabla g(y) - Ax^* + b]^\top \dot{y}.\end{aligned} \tag{B.2}$$

$$\begin{aligned}\dot{V}_2 = &\Big(\frac{2\alpha}{3}t^{\frac{\alpha}{3}-1}(x - x^*) + t^{\frac{\alpha}{3}}\dot{x}\Big)^\top \times \Big(t^{\frac{\alpha}{3}}\ddot{x} + \alpha t^{\frac{\alpha}{3}-1}\dot{x}\\ &+\frac{2\alpha(\alpha-3)}{9}t^{\frac{\alpha}{3}-2}(x - x^*)\Big)\\ &-\frac{2\alpha(\alpha-3)^2}{27}t^{\frac{2\alpha-9}{3}}\|x - x^*\|^2\\ &+\frac{2(3-\alpha)\alpha}{9}t^{\frac{2\alpha}{3}-2}(x - x^*)^{\mathrm{T}}\dot{x},\end{aligned} \tag{B.3}$$

$$\begin{aligned}\dot{V}_3 = &\Big(\frac{2\alpha}{3}t^{\frac{\alpha}{3}-1}(y - y^*) + t^{\frac{\alpha}{3}}\dot{y}\Big)^\top \times \Big(t^{\frac{\alpha}{3}}\ddot{y} + \alpha t^{\frac{\alpha}{3}-1}\dot{y}\\ &+\frac{2\alpha(\alpha-3)}{9}t^{\frac{\alpha}{3}-2}(y - y^*)\Big)\\ &-\frac{2\alpha(\alpha-3)^2}{27}t^{\frac{2\alpha-9}{3}}\|y - y^*\|^2\\ &+\frac{2(3-\alpha)\alpha}{9}t^{\frac{2\alpha}{3}-2}(y - y^*)^{\mathrm{T}}\dot{y}.\end{aligned} \tag{B.4}$$

Plugging (6) and (9) in (B.3) and (B.4), we have

$$\begin{aligned}\dot{V}_2 = &\frac{2\alpha(\alpha^2-9)}{27}t^{\frac{2\alpha}{3}-3}(x - x^*)^2\\ &-\frac{2\alpha}{3}t^{\frac{2\alpha}{3}-1}(x - x^*)^\top(\nabla f(x) - \nabla f(x^*))\\ &-\frac{2\alpha}{3}t^{\frac{2\alpha}{3}-1}(x - x^*)^\top A^\top(y - y^*)\\ &-t^{\frac{2\alpha}{3}}(x - x^*)^\top A^\top \dot{y} - t^{\frac{2\alpha}{3}}(\nabla f(x) - \nabla f(x^*))^\top \dot{x}\\ &-t^{\frac{2\alpha}{3}}\dot{x}^\top A^\top(y - y^*) - \frac{3}{2\alpha}t^{\frac{2\alpha}{3}+1}\dot{x}^\top A^\top \dot{y},\end{aligned} \tag{B.5}$$

$$\begin{aligned}\dot{V}_3 = &\frac{2\alpha(\alpha^2-9)}{27}t^{\frac{2\alpha}{3}-3}(y - y^*)^2\\ &-\frac{2\alpha}{3}t^{\frac{2\alpha}{3}-1}(y - y^*)^\top(\nabla g(y) - \nabla g(y^*))\\ &+\frac{2\alpha}{3}t^{\frac{2\alpha}{3}-1}(y - y^*)^\top A(x - x^*) + t^{\frac{2\alpha}{3}}(y - y^*)^\top A\dot{x}\\ &-t^{\frac{2\alpha}{3}}(\nabla g(y) - \nabla g(y^*))^\top \dot{y}\\ &+t^{\frac{2\alpha}{3}}\dot{y}^\top A(x - x^*) + \frac{3}{2\alpha}t^{\frac{2\alpha}{3}+1}\dot{y}^\top A\dot{x}.\end{aligned} \tag{B.6}$$

Summing (B.2), (B.5), and (B.6) and simplifying the items, we have

$$\begin{aligned}\dot{V} \leq &\frac{2\alpha}{3}t^{\frac{2\alpha}{3}-1}[f(x) - f(x^*) - (x - x^*)^\top \nabla f(x)]\\ &+\frac{2\alpha}{3}t^{\frac{2\alpha}{3}-1}[g(y) - g(y^*) - (y - y^*)^\top \nabla g(y)]\\ &+\frac{2}{27}\alpha(\alpha^2-9)t^{\frac{2\alpha}{3}-3}[(x - x^*)^2 + (y - y^*)^2].\end{aligned}$$

Recall that f and g are convex functions. We have $f(x) - f(x^*) - (x - x^*)^\top \nabla f(x) \leq 0$ and $g(y) - g(y^*) - (y - y^*)^\top \nabla g(y) \leq 0$. Since $0 < \alpha < 3$, $\dot{V} \leq \frac{2}{27}\alpha(\alpha^2-9)t^{\frac{2\alpha}{3}-3}[(x - x^*)^2 + (y - y^*)^2] \leq 0$ and $V(t, x(t), y(t), \dot{x}(t), \dot{y}(t)) \leq V(t_0, x_0, y_0, \dot{x}_0, \dot{y}_0)$.

Define $m_0 = V(t_0, x_0, y_0, \dot{x}_0, \dot{y}_0)$. Because

$$0 \leq V_1(t, x(t), y(t)) \leq V(t, x(t), y(t), \dot{x}(t), \dot{y}(t)) \leq m_0,$$

it follows that

$$L(x(t), y^*) - L(x^*, y(t)) = V_1(t, x(t), y(t))t^{-\frac{2\alpha}{3}} \leq m_0 t^{-\frac{2\alpha}{3}}.$$

(ii) By rearranging terms in V_2 and V_3, we rewrite functions V_2 and V_3 as

$$\begin{aligned}V_2 = &\frac{\alpha^2+3\alpha}{9}t^{\frac{2\alpha}{3}-2}\|x - x^*\|^2 + \frac{1}{2}t^{\frac{2\alpha}{3}}\|\dot{x}\|^2\\ &+\frac{2\alpha}{3}t^{\frac{2\alpha}{3}-1}(x - x^*)^{\mathrm{T}}\dot{x}\\ = &\left\|\frac{\sqrt{\alpha^2+3\alpha}}{3}t^{\frac{\alpha}{3}-1}(x - x^*) + \sqrt{\frac{\alpha}{\alpha+3}}t^{\frac{\alpha}{3}}\dot{x}\right\|^2\\ &+\frac{3-\alpha}{2(\alpha+3)}t^{\frac{2\alpha}{3}}\|\dot{x}\|^2.\end{aligned}$$

It follows that

$$\begin{aligned}\|\dot{x}\|^2 &\leq \frac{2(\alpha+3)}{3-\alpha}\frac{1}{t^{\frac{2\alpha}{3}}}V_2(t, x, \dot{x})\\ &\leq \frac{2(\alpha+3)}{3-\alpha}\frac{1}{t^{\frac{2\alpha}{3}}}V(t_0, x_0, y_0, \dot{x}_0, \dot{y}_0).\end{aligned}$$

Similarly, $\|\dot{y}\|^2 \leq \frac{2(\alpha+3)}{3-\alpha}\frac{1}{t^{\frac{2\alpha}{3}}}V(t_0, x_0, y_0, \dot{x}_0, \dot{y}_0)$.