23rd International Symposium on Mathematical Theory of Networks and Systems
Hong Kong University of Science and Technology, Hong Kong, July 16-20, 2018

Smooth Algorithm For Nonsmooth Convex Optimization Using Proximal Operator and Derivative Feedback

Xianlin Zeng and Jie Chen

***Abstract*— This paper considers a class of nonsmooth convex optimization problems whose objective function is the summation of a twice differentiable convex function and a lower semi-continuous convex function. Using the proximal operator and derivative feedback ideas, we propose a smooth algorithm for the problem and prove the convergence and correctness of the algorithm. The design is then applied to distributed optimization problems.**

***Index Terms*— Nonsmooth convex optimization, proximal operator, derivative feedback, distributed optimization, continuous-time algorithm.**

I. Introduction

Nonsmooth convex optimization problems (see [1]) are very important due to the applications in a wide range of disciplines, such as compressed sensing [2], data fitting [3], and portfolio optimization [4]. Gradient/subgradient methods (see [5]), which are simple to implement and applicable to large-scale problems, are widely studied in a variety of results such as subgradient methods, second order algorithms, and accelerated algorithms with both discrete- and continuous-time algorithms. However, in a nonsmooth convex optimization problem, the analysis of nonsmooth algorithms often involves nonsmooth analysis and is quite challenging and difficult. Therefore, a smooth algorithm for nonsmooth problems is quite appealing for researchers due to the ease of its analysis and implementation.

Proximal methods (see [6]) are a higher level of abstraction of classical optimization algorithms and the proximal algorithm of a nonsmooth optimization problem may be smooth. Like Newton's method is a standard tool for unconstrained smooth optimization problems of small or modest size, proximal algorithms are a useful tool for nonsmooth, constrained, large-scale, or distributed optimization problems. Proximal methods are applicable and especially well-suited to problems involving big data and large-dimensional problems (see [6], [7]).

Recently, continuous-time algorithms for optimization problems have become increasingly important. On one hand, the design and analysis of continuous-time algorithms are often easier than discrete-time ones due to the well-development of stability theory. On the other hands, the discrete versions of continuous-time algorithms are also useful as long as a proper step size is chosen. Thus, continuous-time algorithms are often viewed as prototypes of discrete-time algorithms. Additionally, in some tasks, the optimization is conducted by physical systems or hardware, which have continuous-time evolutions.

Many modern optimization problems arise in network design and operation, finance, supply chain management, scheduling, and many other areas. As a result, distributed optimization algorithms have attracted a significant amount of attention (see [8]–[13]). However, the proposed algorithms for nonsmooth convex optimization problems are often nonsmooth and difficult to analyze due to the noosmoothness. Therefore, distributed smooth algorithms for nonsmooth optimization problems have advantages over nonsmooth algorithms on the analysis and proximal operators may be a useful tool to achieve this goal.

The focus of this paper is to propose a smooth continuous-time algorithm for nonsmooth convex optimization problems using ideas from control and optimization. The contributions of the paper are summarized as follows. Firstly, we present a smooth continuous-time algorithm using the proximal operator and derivative feedback, which gives a new idea for nonsmooth optimization problems. Secondly, the convergence and correctness of the proposed algorithm are proved using stability theory, which provides novel insights into analysis of primal-dual type algorithms. Finally, the design method has potential applications in the area of distributed optimization.

The remainder of the paper is organized as follows. In Section II, the notation, some basic mathematical definitions, and theoretical results are presented. In Section III, a nonsmooth convex optimization problem is formulated, a smooth continuous-time algorithm using the proximal operator and derivative feedback ideas is proposed, and the proof for the correctness of the algorithm is presented. In Section IV, we apply our theoretical results to distributed optimization problems, which gives a new idea for designing distributed algorithms. Finally, Section V concludes this paper.

II. Mathematical Preliminaries

Specifically, $\mathbb{R}$ denotes the set of 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

X. Zeng (xianlin.zeng@bit.edu.cn) is with Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology, Beijing, 100081, China.

J. Chen (chenjie@bit.edu.cn) is with Beijing Advanced Innovation Center for Intelligent Robots and Systems (Beijing Institute of Technology), Key Laboratory of Biomimetic Robots and Systems (Beijing Institute of Technology), Ministry of Education, Beijing, 100081, China.

This work was supported by Projects of Major International (Regional) Joint Research Program NSFC (Grant no. 61720106011), NSFC (Grant no. 61621063, 61781340258, 61603378), China Postdoctoral Science Foundation (2017M620020).

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); $\overline{\mathcal{S}}$ denotes the closure of the subset $\mathcal{S} \subset \mathbb{R}^n$; $\mathrm{int}(\mathcal{S})$ denotes the interior of the subset $\mathcal{S}$; $\dim(\mathcal{S})$ denotes the dimension of the vector space $\mathcal{S}$; $\mathcal{B}_\epsilon(\alpha), \alpha \in \mathbb{R}^n, \epsilon > 0$, denotes the open ball *centered* at α with *radius* ϵ; $\mathrm{dist}(p, \mathcal{M})$ denotes 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 be a lower semi-continuous convex function. The proximal operator of f is

$$\mathrm{prox}_f(v) = \arg\min_x f(x) + \frac{1}{2}\|x - v\|^2.$$

Define the indicator function of a closed convex set Ω as $I_\Omega(x) = 0$ if $x \in \Omega$ and $I_\Omega(x) = +\infty$ otherwise. We have $\mathrm{prox}_{I_\Omega}(v) = P_\Omega(v)$, where $P_\Omega(v) = \arg\min_{x\in\Omega} \|x - v\|$ is the projection operator. Let $\partial f(x)$ denote the subgradient of $f(\cdot)$ at x. Then $\partial f(x)$ is monotone, that is, $(p_x - p_y)^{\mathrm{T}}(x - y) \geq 0$ for all x, y, $p_x \in \partial f(x)$, and $p_y \in \partial f(y)$. $x = \mathrm{prox}_f(v)$ is equivalent to

$$v - x \in \partial f(x). \tag{1}$$

Consider a system

$$\dot{x}(t) = \phi(x(t)), \quad x(0) = x_0, \quad t \geq 0, \tag{2}$$

where $\phi : \mathbb{R}^q \to \mathbb{R}^q$ is Lipschitz continuous. The following result is a special case of [14, Theorem 3.1].

Lemma 2.1: Let $\mathcal{D}$ be a compact, positive invariant set with respect to system (2), $V : \mathbb{R}^q \to \mathbb{R}$ be a continuously differentiable function, and $x(\cdot) \in \mathbb{R}^q$ be a solution of (2) with $x(0) = x_0 \in \mathcal{D}$. Assume $\dot{V}(x) \leq 0, \forall x \in \mathcal{D}$, and define $\mathcal{Z} = \{x \in \mathcal{D} : \dot{V}(x) = 0\}$. If every point in the largest invariant subset $\mathcal{M}$ of $\overline{\mathcal{Z}} \bigcap \mathcal{D}$ is Lyapunov stable, where $\overline{\mathcal{Z}}$ is the closure of $\mathcal{Z} \subset \mathbb{R}^n$, then (2) converges to one of its equilibrium point.

III. Optimization Algorithm Design

A. Problem Description

Consider an optimization problem

$$\min_x f(x) + g(x),\ Ax - b = 0, \tag{3}$$

where $x \in \mathbb{R}^q$, $A \in \mathbb{R}^{m\times q}$, and $b \in \mathbb{R}^m$, $f : \mathbb{R}^q \to \mathbb{R}$ is a twice differentiable convex function, and $g : \mathbb{R}^q \to \mathbb{R}$ is a proper convex closed function.

Remark 3.1: This problem is a very general model. If $g(\cdot)$ is an indicator function of a convex set Ω, then $g(\cdot)$ is equivalent to a set constraint $x \in \Omega$; if $g(x) = \gamma|x|_1$ and $f(x)$ is of a quadratic form, the optimization problem is a LASSO problem. ♦

Then, we arrive at the following lemma by the KKT condition of convex optimization problems.

Lemma 3.1: Suppose problem (3) satisfies Slater's condition. A feasible point $x^* \in \mathbb{R}^q$ is a minimizer to problem (3) if and only if there exists $\lambda^* \in \mathbb{R}^m$ such that

$$0_q \in -\nabla f(x^*) - \partial g(x^*) - A^{\mathrm{T}}\lambda^*, \tag{4}$$
$$0_m = Ax^* - b. \tag{5}$$

B. Optimization Algorithm

To solve problem (3), we proposed a dynamical system as

$$\dot{x}(t) = \mathrm{prox}_g\Big[x(t) - \nabla f(x(t)) - A^{\mathrm{T}}\lambda(t)\Big] - x(t), \tag{6a}$$
$$\dot{\lambda}(t) = A(x(t) + \dot{x}(t)) - b, \tag{6b}$$

where $t \geq 0$, $x(0) = x_0 \in \mathbb{R}^q$, and $\lambda(0) = \lambda_0 \in \mathbb{R}^m$.

Algorithm (6) uses the proximal method and derivative feedback. It is a primal-dual method to solve the saddle points of the Lagrangian function $L(x, \lambda) = f(x) + g(x) + \lambda^{\mathrm{T}}(Ax - b)$. This algorithm has two good properties:

- **Convergence:** Note that the saddle point dynamics $\dot{x} = -\nabla_x L(x, \lambda)$ and $\dot{\lambda} = \nabla_\lambda L(x, \lambda)$ are not convergent in general. The derivative feedback design plays as a damping part to make the algorithm convergent and the proof will be shown in the convergence analysis.
- **Smoothness:** Because the proximal operator $\mathrm{prox}_g(\cdot)$ is continuous and nonexpansive, the proposed algorithm is a Lipschitz continuous system, even though problem (3) is a nonsmooth problem.

Lemma 3.2: Suppose problem (3) satisfies Slater's condition. (x^*, λ^*) is an equilibrium of algorithm (6) if and only if x^* is a solution to problem (3).

Proof: By (1), $\mathrm{prox}_g\Big[x^* - \nabla f(x^*) - A^{\mathrm{T}}\lambda^*\Big] = x^*$ if and only if (4) holds. Hence, (x^*, λ^*) is an equilibrium of algorithm (6) if and only if (4) and (5) are satisfied. By Lemma 3.1, the conclusion is obtained ■

C. Convergence Analysis

In this subsection, we state the convergence result.

Theorem 3.1: Assume problem (3) has a solution and satisfies Slater's condition.

(i) every equilibrium of (6) is Lyapunov stable and every solution $(x(t), \lambda(t))$ is bounded;
(ii) the trajectory $(x(t), \lambda(t))$ converges and $\lim_{t\to\infty} x(t)$ is a solution to problem (3).

Proof: (i) If problem (3) has a solution, it follows from Lemma 3.2 that algorithm (6) has an equilibrium. Let (x^*, λ^*) be an equilibrium of algorithm (6). By (6a), we have

$$\dot{x} + x = \mathrm{prox}_g\Big[x - \nabla f(x) - A^{\mathrm{T}}\lambda\Big].$$

It follows from (1) that

$$-\nabla f(x) - A^{\mathrm{T}}\lambda - \dot{x} \in \partial g(\dot{x} + x). \tag{7}$$

Similarly, we have

$$-\nabla f(x^*) - A^{\mathrm{T}}\lambda^* \in \partial g(x^*). \tag{8}$$

Because $g(\cdot)$ is a convex function, $\partial g(x)$ is monotone and

$$(-\nabla f(x) - A^{\mathrm{T}}\lambda - \dot{x} + \nabla f(x^*) + A^{\mathrm{T}}\lambda^*)^{\mathrm{T}} \cdot(\dot{x} + x - x^*) \geq 0.$$

One can obtain that

$$\begin{aligned}\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}(x-x^*)^2 + \frac{\mathrm{d}}{\mathrm{d}t}[f(x) - f(x^*) - (x-x^*)^{\mathrm{T}}\nabla f(x^*)]\\ = \;& \dot{x}^{\mathrm{T}}(x-x^*) + \dot{x}^{\mathrm{T}}(\nabla f(x) - \nabla f(x^*))\\ \leq \;& -\dot{x}^2 - \dot{x}^{\mathrm{T}}A^{\mathrm{T}}(\lambda-\lambda^*)\\ & -(x-x^*)^{\mathrm{T}}(\nabla f(x) - \nabla f(x^*))\\ & -(x-x^*)A^{\mathrm{T}}(\lambda-\lambda^*). \end{aligned} \tag{9}$$

It follows from (6b) that

$$\begin{aligned}\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}(\lambda-\lambda^*)^2 &= \dot{\lambda}^{\mathrm{T}}(\lambda-\lambda^*)\\ &= (\lambda-\lambda^*)^{\mathrm{T}}(A(x+\dot{x})-b). \end{aligned} \tag{10}$$

Define $V(x,\lambda) = f(x) - f(x^*) - (x-x^*)^{\mathrm{T}}\nabla f(x^*) + \frac{1}{2}(x-x^*)^2 + \frac{1}{2}(\lambda-\lambda^*)^2$ and note that $Ax^* = b$. In view of (9) and (10), we have

$$\dot{V}(x,\lambda) \leq -\dot{x}^2 - (x-x^*)^{\mathrm{T}}(\nabla f(x) - \nabla f(x^*)).$$

Because $f(\cdot)$ is convex, $\dot{V}(x,\lambda) \leq -\dot{x}^2 \leq 0$. Additionally, $f(x) - f(x^*) - (x-x^*)^{\mathrm{T}}\nabla f(x^*) \geq 0$, and hence, $V(x,\lambda) \geq \frac{1}{2}(x-x^*)^2 + \frac{1}{2}(\lambda-\lambda^*)^2$. As a result, $V(x,\lambda)$ is positive-definite, radically unbounded, lower bounded. Hence, (x^*,λ^*) is Lyapunov stable and the trajectory $(x(t),\lambda(t))$ is bounded.

(ii) Define

$$\mathcal{R} = \{(x,\lambda) : 0 = \dot{V}(x,\lambda)\} \subset \{(x,\lambda) : \dot{x} = 0_q\}.$$

Let $\mathcal{M}$ be the largest invariant set of $\mathcal{R}$. It follows from the invariance principle (Theorem 2.41 of [15]) that $(x(t),\lambda(t)) \to \mathcal{M}$ as $t \to \infty$ and $\mathcal{M}$ is positive invariant. Assume $(\overline{x}(t),\overline{\lambda}(t))$ is a trajectory of (6) such that $(\overline{x}(0),\overline{\lambda}(0)) \in \mathcal{M}$. Then $(\overline{x}(t),\overline{\lambda}(t)) \in \mathcal{M}$ for all $t \geq 0$. Therefore, $\dot{\overline{x}}(t) \equiv 0_q$ and

$$\dot{\overline{\lambda}}(t) \equiv A\overline{x}(0) - b.$$

If $\dot{\overline{\lambda}}(t) \neq 0_m$, then $\overline{\lambda}(t)$ becomes unbounded, which is a contradiction to part (i). Hence, $\dot{\overline{\lambda}}(t) = 0_m$ and $\mathcal{M} \subset \{(x,\lambda) : \dot{x} = 0_q,\ \dot{\lambda} = 0_m\}$. By part (i), every point in $\mathcal{M}$ is Lyapunov stable. It follows from Lemma 2.1 that $(x(t),\lambda(t))$ converges to an equilibrium point. Due to Lemma 3.2, $\lim_{t\to\infty} x(t)$ is an solution to problem (3). ■

Remark 3.2: Theorem 3.1 shows the convergence property of algorithm (6) in this paper. Part (i) of Theorem 3.1 shows that every equilibrium of the algorithm is Lyapunov stable and the state trajectories of the algorithm are bounded; part (ii) of Theorem 3.1 further proves that the state trajectory converges to an equilibrium point. ♦

IV. Application to Distributed Optimization

In this section, we apply our proposed algorithm to a class of distributed optimization problems.

Some additional notation in graph theory is needed in this section. A weighted 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, $A = [a_{i,j}] \in \mathbb{R}^{n\times n}$ is a *weighted 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 weighted graph $\mathcal{G}$ is undirected and connected, then $L_n = L_n^{\mathrm{T}} \geq 0$, $\operatorname{rank} L_n = n-1$ and $\ker(L_n) = \{k\mathbf{1}_n : k \in \mathbb{R}\}$.

Consider a network of n agents interacting over a graph $\mathcal{G}$. Then a distributed algorithm is need to solve

$$\min_{x\in\mathbb{R}^{nq}} \sum_{i=1}^n f^i(x_i) + g^i(x_i), \tag{11a}$$

$$s.t.\ x_i = x_j,\ \forall i,j \in \{1,\ldots,n\}, \tag{11b}$$

where $x = [x_1^{\mathrm{T}},\cdots,x_n^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{nq}$, agent i uses only its own local data x_i and exchanged information with its neighbors.

Remark 4.1: Problem (11) is a model that has vast applications. If $g^i(x_i)$ is a regularized function, the problem is a very widely used model in machine learning such as l_1 regularized loss minimization and LASSO. The special case, in which $g^i(x_i) = I_{\Omega_i}(x_i)$, has been widely investigated in [8], [10], [13], [16], [17]. ♦

Assumption 4.1: To ensure the wellposedness of the problem and algorithm, it is assumed that

1) f^i is twice differentiable continuous convex for all $i \in \{1,\ldots,n\}$,
2) g^i is (nonsmooth) lower semi-continuous convex for all $i \in \{1,\ldots,n\}$, whose proximal operator is assumed to be easy to obtain;
3) the weighted graph $\mathcal{G}$ is connected and undirected.
4) there exists at least one finite solution to problem (11).

Define $x \triangleq [x_1^{\mathrm{T}},\ldots,x_n^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{nq}$ and define $L \triangleq L_n \otimes I_q \in \mathbb{R}^{nq\times nq}$, where $L_n \in \mathbb{R}^{n\times}$ is the Laplacian matrix of $\mathcal{G}$. Let $f(x) = \sum_{i=1}^n f^i(x_i)$ and $g(x) = \sum_{i=1}^n g^i(x_i)$. Clearly, problem (11) is equivalent to the following problem

$$\min_{x\in\mathbb{R}^{nq}} f(x) + g(x), \quad Lx = 0_{nq}.$$

Then we propose the distributed algorithm as

$$\begin{aligned}\dot{x}_i(t) \in &\operatorname{prox}_{g^i}\Big[x_i(t) - \nabla f^i(x_i(t))\\ &- \sum_{j=1}^n a_{i,j}(\lambda_i(t) - \lambda_j(t))\Big] - x_i(t), \end{aligned} \tag{12}$$

$$\dot{\lambda}_i(t) = \sum_{j=1}^n a_{i,j}(x_i(t) + \dot{x}_i(t) - x_j(t) - \dot{x}_j(t)), \tag{13}$$

where $t \geq 0$, $x_i(0) = x_{i0} \in \Omega_i \subset \mathbb{R}^q$, $\lambda_i(0) = \lambda_{i0} \in \mathbb{R}^q$, and $a_{i,j}$ is the (i,j)th element of the weighted adjacency matrix of graph $\mathcal{G}$. Define $\lambda \triangleq [\lambda_1^{\mathrm{T}}, \ldots, \lambda_n^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{nq}$. The compact form of algorithm (12) is

$$\dot{x}(t) = \mathrm{prox}_g\Big[x(t) - \nabla f(x(t)) - L\lambda(t)\Big] - x(t), \quad (14a)$$

$$\dot{\lambda}(t) = L(x(t) + \dot{x}(t)), \quad (14b)$$

The correctness and convergence of algorithm (14) can be obtained following Lemma 3.2 and Theorem 3.1.

Remark 4.2: Using the proximal operator and derivative feedback design, Algorithm (14) is a smooth algorithm and its analysis is easier compared with nonsmooth algorithms in [8], [10], [13]. This framework gives new ideas for designing distributed algorithms for nonsmooth optimizations without using nonsmooth analysis. ♦

V. Conclusion

This paper has proposed a smooth continuous-time algorithm that solves a class of nonsmooth convex optimization problems. Using the proximal operator and derivative feedback design, the proposed distributed algorithm is able to solve the optimization problem with any initial value. The convergence property of the algorithm has been analyzed via the stability theory and the results have been applied to a distributed nonsmooth optimization problem.

References

[1] Y. Nesterov, *Nonsmooth Convex Optimization. In: Introductory Lectures on Convex Optimization*. Boston, MA.: Springer, 2004, vol. 87.

[2] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] J. Xiao, Y. Yang, and X. Yuan, "On l_2 data ftting and modified nonconvex nonsmooth regularization for image recovery," *Journal of Computational Analysis & Applications*, vol. 15, no. 1, pp. 264–279, 2013.

[4] M. Fakhar, M. R. Mahyarinia, and J. Zafaran, "On nonsmooth robust multiobjective optimization under generalized convexity with applications to portfolio optimization," *European Journal of Operational Research*, vol. 265, no. 1, pp. 39–48, 2018.

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

[6] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2014.

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

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

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

[10] X. Zeng, P. Yi, and Y. Hong, "Distributed continuous-time algorithm for constrained convex optimizations via nonsmooth analysis approach," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5227–5233, 2017.

[11] G. Shi, K. Johansson, and Y. Hong, "Reaching an optimal consensus: Dynamical systems that compute intersections of convex sets," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 610–622, 2013.

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

[13] Q. Liu and J. Wang, "A second-order multi-agent network for bound-constrained distributed optimization," *IEEE Transactions on Automatic Control*, vol. 60, no. 12, pp. 3310–3315, 2015.

[14] Q. Hui, W. M. Haddad, and S. P. Bhat, "Semistability, finite-time stability, differential inclusions, and discontinuous dynamical systems having a continuum of equilibria," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2465–2470, 2009.

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

[16] Z. Deng and Y. Hong, "Multi-agent optimization design for autonomous Lagrangian systems," *Unmanned Systems*, vol. 4, no. 1, pp. 5–13, 2016.

[17] Y. Zhang, Z. Deng, and Y. Hong, "Distributed optimal coordination for multiple heterogeneous Euler-Lagrangian systems," *Automatica*, vol. 79, pp. 207–213, 2017.