SCIENCE CHINA
Information Sciences

• LETTER •

August 2020, Vol. 63 189201:1–189201:3
https://doi.org/10.1007/s11432-018-9682-4

Distributed algorithms for solving the convex feasibility problems

Kaihong LU[1], Gangshan JING[1,2] & Long WANG[3*]

[1]*Center for Complex Systems, School of Mechano-electronic Engineering, Xidian University, Xi'an* 710071*, China;*
[2]*Department of Applied Mathematics, Hong Kong Polytechnic University, Hong Kong, China;*
[3]*Center for Systems and Control, College of Engineering, Peking University, Beijing* 100871*, China*

Received 12 April 2018/Revised 26 September 2018/Accepted 31 October 2018/Published online 3 March 2020

Citation Lu K H, Jing G S, Wang L. Distributed algorithms for solving the convex feasibility problems. Sci China Inf Sci, 2020, 63(8): 189201, https://doi.org/10.1007/s11432-018-9682-4

Dear editor,

Along with the recent adoption of multi-agent networks [1–9], consensus-based distributed optimization has been extensively investigated. In [1], a distributed algorithm that combines the consensus and subgradient algorithms was developed for obtaining the solution of the unconstrained convex distributed optimization problems. In [2], a distributed discrete-time projected subgradient algorithm was proposed to solve the constrained optimization problems in which each agent has to remain in its own convex set. In [3], a distributed continuous-time subgradient algorithm was presented to compute the distributed optimizations that exhibited convex inequality constraints. Furthermore, in [4], by considering the linear equation constraints, convex inequality constraints, and convex set constraints, a distributed algorithm was developed based on the consensus algorithm and the saddle point algorithm. The researchers in [1–4] assume that the directed graph is balanced. Various special optimization problems such as computing the intersection of convex sets [2] and solving linear equations [7, 8], have been treated using distributed computations. The convex feasibility problem (CFP) is more general than the computation of the intersection of convex sets. Further, while dealing with a CFP, one should ensure that the solution is obtained in the intersection of the relevant convex sets, and the linear equations or convex inequalities should also be solved.

This study develops a distributed approach to solve CFPs. Based on the subgradient and projection algorithms, both continuous- and discrete-time distributed algorithms are proposed for multi-agent systems. The coexistence of subgradient and projection operations introduces nonlinearity into these algorithms, which introduces difficulties while analyzing the convergence of the proposed algorithm. Unlike the distributed algorithms that are used for solving linear equations [7, 8] in which the initial state of each agent is required to satisfy the equation, CFPs can be solved using our algorithms with arbitrary initial states. While implementing the proposed algorithms, all the states of the agents will reach a common asymptotic point if the directed graph is strongly connected this point is located in the solution set of the CFP. We will prove this statement without a requirement that the directed graph should be balanced. Additionally, the convergence of the proposed algorithm does not rely on a diminishing step size.

A function $f(\cdot) : \mathbb{R}^m \rightarrow \mathbb{R}$ is convex if $f(\gamma x + (1-\gamma)y) \leqslant \gamma f(x)+(1-\gamma)f(y)$ for any $x \neq y \in \mathbb{R}^m$ and $0 < \gamma < 1$. For convex function $f(x)$, if $\langle \nabla f(x), y - x \rangle \leqslant f(y) - f(x)$ for any $y \in \mathbb{R}^m$, then $\nabla f(x)$ is a subgradient of function f at point $x \in \mathbb{R}^m$. Additionally, subgradients must exist for any convex function. Furthermore, if the convex

* Corresponding author (email: longwang@pku.edu.cn)

function is differentiable, its gradient is considered to be a unique subgradient. The set $\Omega \subset \mathbb{R}^m$ is convex if $\gamma x + (1-\gamma)y \in \Omega$ for any scalar $0 < \gamma < 1$ and $x, y \in \Omega$. For a closed convex set Ω, let $\|x\|_\Omega \triangleq \inf_{y\in\Omega} \|x-y\|$ denote the standard Euclidean distance of vector $x \in \mathbb{R}^m$ from Ω. Further, there is a unique element $P_\Omega(x) \in \Omega$ such that $\|x - P_\Omega(x)\| = \|x\|_\Omega$, where $P_\Omega(\cdot)$ can be referred to as the projection onto the set Ω. Furthermore, $\|x - P_\Omega(x)\|^2$ is differentiable with respect to x and $\nabla \|x\|_\Omega^2 = 2(x - P_\Omega(x))$.

Consider a multi-agent system comprising n agents labeled with the set $\mathcal{V} = \{1, \ldots, n\}$. Further, we consider agents with continuous-time dynamics, which can be given as

$$\dot{x}_i(t) = u_i(t), \quad i \in \mathcal{V} \tag{1}$$

and discrete-time dynamics, which can be given as

$$x_i(t+1) = x_i(t) + u_i(t), \quad i \in \mathcal{V}, \tag{2}$$

where $x_i(t) \in \mathbb{R}^m$ and $u_i(t) \in \mathbb{R}^m$ denote the state and input of agent i, respectively. Further, we intend to design $u_i(t)$ for (1) and (2) using only the local information for solving the following CFP:

$$\begin{cases} g_i(x) \leqslant 0, \\ x \in X : \triangleq \cap_{i=1}^n X_i, \end{cases} \quad i = 1, \ldots, n, \tag{3}$$

where $x \in \mathbb{R}^m$, $g_i(\cdot) : \mathbb{R}^m \to \mathbb{R}$ is a convex function. This function is continuous on $(-\infty, \infty)$. Each X_i is a closed convex set. Agent i can access only the information associated with the subgradient $\nabla g_i^+(\cdot)$ and the projection $P_{X_i}(\cdot)$. The solution set of the CFP (3) is denoted by X^*. Further, the following assumption can be adopted throughout the study.

Assumption 1. X^* is non-empty.

Note that $x = P_{X_i}(x)$ if and only if $x \in X_i$. If $x = P_{X_i}(x)$ for all $i = 1, \ldots, n$, then $x \in \cap_{i=1}^n X_i$. Because the algorithms in this study refer to the projection operator $P_{X_i}(\cdot)$, here we only consider the convex sets X_i onto which the projection $P_{X_i}(x)$ can be easily calculated or for which a detailed expression can be given at any point x. For example, if set X represents the solution set of the linear equation $a^{\mathrm{T}}x - b = 0$, i.e., $X = \{x | a^{\mathrm{T}}x - b = 0\}$, where $a, x \in \mathbb{R}^m$, $b \in \mathbb{R}$, one can easily obtain that $P_X(x) = (I - \frac{aa^{\mathrm{T}}}{\|a\|^2})x + \frac{ba}{\|a\|^2}$ is a projection of x onto set X. Consequently, the algorithms presented in this study can also be applied to solve the CFP using linear equations.

To solve the CFP (3), the following control input for the multi-agent system (1) can be proposed:

$$\begin{cases} u_i(t) = \sum_{j\in N_i} a_{ij}(x_j(t) - x_i(t)) + \phi_i(t), \\ \phi_i(t) = -\tau\big([x_i(t) - P_{X_i}(x_i(t))] \\ \qquad + \nabla g_i^+(x_i(t))\big), \end{cases} \quad i \in \mathcal{V}, \tag{4}$$

where τ is the positive coefficient. Note that ϕ_i depends only on the state of agent i so that Eq. (4) is distributed. $\nabla g_i^+(x) = 0$ if $g_i(x) \leqslant 0$; otherwise, $\nabla g_i^+(x) = \nabla g_i(x)$.

Theorem 1. If the directed graph $\mathcal{G}(\mathcal{A})$ is strongly connected, the multi-agent system (1) with (4) reaches consensus asymptotically, and the consensus state is located in set X^*.

For a discrete-time multi-agent system (2), the following input is proposed for solving the CFP (3):

$$\begin{cases} u_i(t) = h\sum_{j\in N_i} a_{ij}(x_j(t) - x_i(t)) + \phi_i(t), \\ y_i(t) = x_i(t) + h\sum_{j\in N_i} a_{ij}(x_j(t) - x_i(t)), \\ \xi_i(t) = \frac{1}{2}\|y_i(t) - P_{X_i}(y_i(t))\|^2 + g_i^+(y_i(t)), \\ \varphi_i(t) = \|y_i(t) - P_{X_i}(y_i(t))\|^2 + \|\nabla g_i^+(t)\|^2, \\ \phi_i(t) = \begin{cases} -\frac{\epsilon\xi_i(t)}{2\varphi_i(t)}(y_i(t) - P_{X_i}(y_i(t)) \\ \quad + \nabla g_i^+(t)), \quad \text{if} \quad \varphi_i(t) \neq 0, \\ 0, \quad \text{otherwise}, \end{cases} \end{cases} \quad i \in \mathcal{V}, \tag{5}$$

where $0 < \epsilon < 2$ and $\nabla g_i^+(x) = 0$ if $g_i(x) \leqslant 0$; otherwise, $\nabla g_i^+(x) = \nabla g_i(x)$. $\nabla g_i^+(t)$ denotes the subgradient of function $g_i^+(y)$ at $y = y_i(t)$ and h is the control gain that has to be designed. Note that each agent can only access information about its own inequality and set as well as its own state and the relations between itself and its immediate neighbors. Therefore, Eq. (5) is distributed.

Theorem 2. If the directed graph $\mathcal{G}(\mathcal{A})$ is strongly connected and $0 < h < \frac{1}{\max_{1\leqslant i\leqslant n}(\sum_{j=1}^n a_{ij})}$, the multi-agent system (2) with (5) reaches consensus asymptotically, and the consensus state is in set X^*.

Simulations. Consider a multi-agent system comprising five agents by intending to cooperatively search a feasibility (solution) $z^* = [z_1^*, z_2^*]^{\mathrm{T}}$ of the CPF that includes two closed convex sets, $X_1 = \{(z_1, z_2) | 2 \leqslant z_1 \leqslant 4, 0 \leqslant z_2 \leqslant 2\}$ and $X_2 = \{(z_1, z_2) | 2.5 \leqslant z_1 \leqslant 4.5, 1 \leqslant z_2 \leqslant 3\}$, and three linear inequalities, $c(z) = 2z_1 - 3z_2 - 2 \leqslant 0$, $d(z) = 2z_1 + 3z_2 - 11 \leqslant 0$, and $q(z) = 8z_1 - 3z_2 - 28 \leqslant 0$. In Figure 1(a), the yellow region represents the feasible region. Set X_i is only known to agent i for

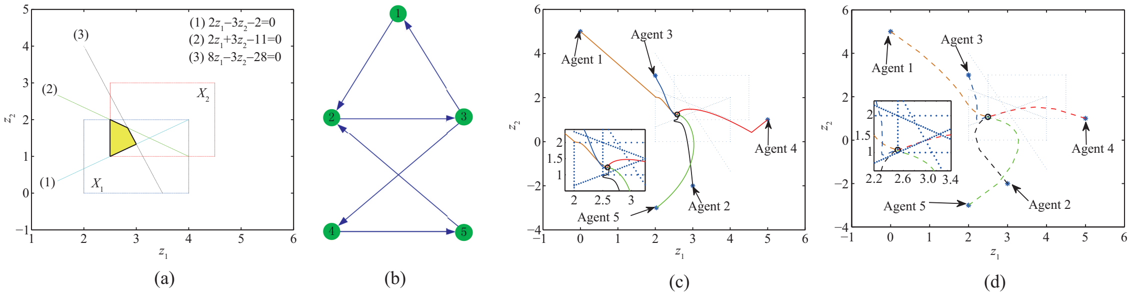

Figure 1 (Color online) (a) Feasible region of the CFP; (b) the communication graph; (c) the trajectory of the multi-agent system in the continuous-time case; (d) the trajectory of the multi-agent system in the discrete-time case. "*" represents the initial states of the agents while "∘" represents their final states.

$i = 1, 2$; further, agents 3–5 can only have access to $c(z), d(z)$, and $q(z)$, respectively. We present the simulation results for continuous- and discrete-time distributed algorithms. The communication graph is directed in each case.

(1) Continuous-time case. The communication graph is depicted in Figure 1(b) and is observed to be strongly connected. The weight of each edge connecting different agents is 1. The coefficient is set to $\tau = 20$, and let the initial state of each agent be $x_1(0) = [0, 5]^{\rm T}, x_2(0) = [3, -2]^{\rm T}, x_3(0) = [2, 3]^{\rm T}, x_4(0) = [5, 1]^{\rm T}, x_5(0) = [2, -3]^{\rm T}$. The trajectory of the multi-agent system (1) using (4) is depicted in Figure 1(c). The agents reach consensus at $z^* = [2.58, 1.23]^{\rm T}$, which is a solution to the CFP. This observation is consistent with that of Theorem 1.

(2) Discrete-time case. The communication graph that is used in the first case is again used in this case. We set $\epsilon = 0.85$ and $h = 0.85$. Under the same initial conditions as those in the previous case, the trajectory of the multi-agent system (2) with (5) is depicted in Figure 1(d). The agents reach consensus at $z^* = [2.59, 1.49]^{\rm T}$, which is a solution to the CFP. This observation is consistent with that of Theorem 2.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61751301, 61533001).

Supporting information Appendixes A and B. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Nedić A, Ozdaglar A. Distributed subgradient methods for multi-agent optimization. IEEE Trans Automat Contr, 2009, 54: 48–61
2 Nedić A, Ozdaglar A, Parrilo P A. Constrained consensus and optimization in multi-agent networks. IEEE Trans Automat Contr, 2010, 55: 922–938
3 Lu K, Jing G, Wang L. Distributed algorithm for solving convex inequalities. IEEE Trans Automat Contr, 2018, 63: 2670–2677
4 Zhu M H, Martínez S. On distributed convex optimization under inequality and equality constraints. IEEE Trans Automat Contr, 2012, 57: 151–164
5 Wang L, Shi H, Chu T G, et al. Aggregation of foraging swarms. In: Advances in Artificial Intelligence. Berlin: Springer, 2004. 3339: 766–777
6 Wang L, Xiao F. Finite-time consensus problems for networks of dynamic agents. IEEE Trans Automat Contr, 2010, 55: 950–955
7 Mou S, Liu J, Morse A S. A distributed algorithm for solving a linear algebraic equation. IEEE Trans Automat Contr, 2015, 60: 2863–2878
8 Cao H, Gibson T E, Mou S, et al. Impacts of network topology on the performance of a distributed algorithm solving linear equations. In: Prcoeedings of 2016 IEEE 55th Conference on Decision and Control (CDC). New York: IEEE, 2016. 1733–1738
9 Wang L, Xiao F. A new approach to consensus problems in discrete-time multiagent systems with time-delays. Sci China Ser F-Inf Sci, 2007, 50: 625–635