A Distributed Algorithm for Resource Allocation Over Dynamic Digraphs

Yun Xu, Tingrui Han, Kai Cai, Zhiyun Lin, *Senior Member, IEEE*, Gangfeng Yan, and Minyue Fu, *Fellow, IEEE*

***Abstract*—This paper studies a distributed resource allocation problem for a multiagent network with a time-varying digraph. Each agent in the network is associated with a local variable (resource) and a convex cost function. The goal is to collectively minimize the total cost in a distributed fashion, subject to individual resource constraints, and collective equality constraints. The main challenge of the problem is due to the local information structure imposed by the time-varying digraph that should be considered as part of the problem formulation. This paper develops a nonnegative surplus-based distributed optimization algorithm. It is shown that the proposed distributed algorithm converges to the global minimizer provided that the time-varying digraph is jointly strongly connected. Also, all the parameters used in the proposed algorithm rely only on local knowledge.**

***Index Terms*—Distributed optimization, resource allocation, multi-agent systems.**

I. Introduction

THE resource allocation problem deals with how to allocate available resources to a number of users, called agents. In this paper, we deal with the multiple resource allocation problem modelling a collection of independent resources. That is, considering a network of n agents, for each agent $i \in \{1, \cdots, n\}$, we associate a variable $x_i \in \mathbb{R}$ and a corresponding cost function $F_i(x_i)$. The resource allocation problem is the optimization problem aiming to find optimal x_i to $\underset{x_1,\ldots,x_n}{\text{minimize}} \quad \sum_{i=1}^{n} F_i(x_i)$ under the collective equality constraints $Cx = d$ and individual inequality (state) constraints $\underline{x}_i \leq x_i \leq \overline{x}_i$, where $\underline{x}_i$ and $\overline{x}_i$ are given constants, d is a given vector, and C is a non-negative matrix with each column having only one non-zero entry. This problem can address applications where multiple resources are to be allocated, for example, in energy management systems two resources (generation and demand) are typically allocated [1], [2]). This problem can also model the special case where there is only one resource to be allocated, e.g., economic dispatch problems [3], power regulation [4] and take-or-pay fuel supply problems [5].

Although many centralized optimization algorithms exist for this problem, they are not suitable for large-scale networks due to significant communications and computational overhead required for collecting information from all the agents in the network. Additional disadvantages of the centralized approach include required global knowledge of the whole network, lack of robustness due to time-varying nature of the network, and lack of privacy (by requiring individual agents to provide information) [3]. These disadvantages motivate the need of study for distributed resource allocation problems.

In order to characterize the network constraints for information exchanging in distributed resource allocation problems, graphs are usually adopted to describe inter-agent communication topologies. The undirected graph model is considered in [6]–[10], which means that each agent communicates with its neighbors in a bidirectional manner. However, as in many applications (e.g., in smart grid [11]) using wireless networks, communications between some agents in the network may be unidirectional due to the use of directional antennas or due to the heterogeneous nature of the wireless communication nodes [12]. Besides, there are inevitable factors leading to directed communication due to packet loss and communication interference in wireless networks. In energy related applications, considering directed communication can also strengthen the scalability of distributed energy management [1]. Therefore, a directed network is unavoidable [3], [13], [14]. Moreover, as the communication links between some nodes may become disconnected sometimes due to external interference, a time-varying directed graph model is a more practical one, which includes the static undirected graph model as a special one. In other words, distributed algorithms need to be developed, which should work for time-varying directed networks rather than only for static undirected networks.

Distributed algorithms have been studied to solve the resource allocation problems since the center-free algorithms firstly proposed in [15]. The flow control problem has a similar optimization problem formulation to the resource allocation problem.

Manuscript received May 8, 2016; revised November 3, 2016 and January 19, 2017; accepted January 30, 2017. Date of publication February 15, 2017; date of current version March 13, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ami Wiesel. The work of K. Cai was supported in part by Program to Disseminate Tenure Tracking System and Grants-in-Aid for Scientific Research 26870169, MEXT, Japan, and in part by the Open Research Project (ICT1600183) of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China. The work of Z. Lin was supported by the National Natural Science Foundation of China 61673344. *(Corresponding author: Zhiyun Lin.)*

Y. Xu, T. Han, and G. Yan are with the College of Electrical Engineering, Zhejiang University, Hangzhou, Zhejiang 310007, China (e-mail: zxcvbnmxuyun@sina.com; hantingrui@zju.edu.cn; ygf@zju.edu.cn).

K. Cai is with Urban Research Plaza, Osaka City University, Osaka 558-8585, Japan (e-mail: kai.cai@eng.osaka-cu.ac.jp).

Z. Lin is with the School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China and also with the State Key Laboratory of Industrial Control Technology, College of Electrical Engineering, Zhejiang University, Hangzhou 310007, China (e-mail: linz@zju.edu.cn).

M. Fu is with the School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, NSW 2308, Australia, and also with the Zhejiang University, Hangzhou 310007, China (e-mail: minyue.fu@newcastle.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2017.2669896

However, the algorithms in [16]–[18], developed to solve the flow control problem, involve calculation of a global quantity at each iteration, and therefore may not be suitable for large, scalable networks. As a special type of resource allocation problems, the communication graph and the underlying flow graph are the same in flow control, i.e., the collective equality constraints are related to the communication graph. Different from flow control, we are more interested in the resource allocation problem, for which the communication graph and the collective equality constraints are separate in this paper. For the resource allocation problem without individual state constraints, distributed algorithms are studied under static undirected graphs in [6], [19] and under time-varying undirected graphs in [20], respectively. For the resource allocation problem with individual state constraints, consensus based distributed algorithms are proposed for undirected graphs [7], [8] and balanced digraphs [21]. However, for a directed network, it usually does not satisfy the balanced property, which makes these algorithms not applicable. More recently, a distributed bisection algorithm [14], a ratio consensus based decentralized algorithm [13], and a surplus based consensus algorithm [3] are developed to handle such situations. However, global knowledge, which usually cannot be known in a distributed setting, is required for the algorithms or for the design of some critical parameters in these algorithms. In [13], each agent needs to know the cost function parameters of all other agents, while in [3] a critical parameter named the learning gain can be designed only after knowing some global information of the network and is supposed to be sufficiently small. Moreover, the algorithms presented in [3], [13], [14] assume static topology, which makes them not applicable when the network becomes time-varying due to link losses. For a time-varying digraph, the main difficulty lies in the aspect that information exchanging is dynamic and may be unidirectional for some pairs of nodes, thus rendering distributed algorithms hard to be designed with provable convergence properties.

In the optimization literature, there are similar optimization problems as the resource allocation problem, which aim to $\underset{x_1,\ldots,x_n}{\text{minimize}} \quad \sum_{i=1}^n F_i(x_i)$ subject to coupled inequality/equality constraints [22]–[24]. However, again, balanced digraphs are assumed in [22] and undirected graphs are assumed in [23], [24]. Apart from these optimization problems with the individual cost function $F_i(x_i)$ depending on individual variable x_i, there are also optimization problems (i.e., $\underset{x_1,\ldots,x_d}{\text{minimize}} \quad \sum_{i=1}^n F_i(x)$) with the individual cost function $F_i(x)$ depending on global variable x. For this setup without state constraints, distributed subgradient methods are developed in [25] and [26] over time-varying undirected graphs and static balanced graphs, respectively. By taking into consideration of state constraints, distributed optimization algorithms have been studied in [27] and [28] for time-varying balanced graphs and for static undirected graphs. So far, most works are concerned with distributed optimization problems in undirected and/or balanced multi-agent networks with only a few exceptions [29], [30] that consider directed and time-varying directed multi-agent networks. Since the ideas for solving the resource allocation problem are closely related to a dual version of distributed gradient descent [3], [13], [14], the distributed subgradient methods in [29], [30] are helpful for solving the resource allocation problem with proper modifications. However, each node needs to know some global information of constraints. Moreover, high computation and communication costs may be incurred by these modifications.

Considering the absence of a distributed algorithm for the resource allocation problem in time-varying and directed multi-agent networks, this paper aims to solve the distributed resource allocation problem over time-varying directed networks. Based on a surplus idea for consensus, a fully distributed algorithm is proposed to solve the distributed resource allocation problem, for which the time-varying directed topology is also a part of the constraints to the problem. The meaning of a fully distributed algorithm is two-fold. First, the design of the parameters used in each iteration for each agent is only based on local knowledge available to it. Second, the iteration is carried out by each agent using only information transmitted from its neighbors. We then show that the distributed algorithm ensures global convergence to the optimal resource allocation solution. The idea is inspired by the surplus based averaging algorithm over digraphs [31]. But the fundamental difference is that the Lyapunov function based analysis developed in [31] is no longer applicable for the convergence analysis of our algorithm due to individual state constraints. This paper develops a new technique based on monotonicity analysis and shows that the convergence of our algorithm is guaranteed provided that the time-varying digraph is jointly strongly connected. Compared with these existing distributed algorithms for the resource allocation problem, our work removes the need of doubly stochastic matrices and provides provable convergence results for time-varying and directed multi-agent networks.

This paper is organized as follows. In Section II, preliminaries and problem formulation are introduced. In Section III, a nonnegative-surplus based distributed algorithm for solving the resource allocation problem is designed. Convergence analysis is given in Section IV. Feasible analysis and initialization are discussed in V. At last, simulation results are presented in Section VI and we summarize our paper and state further research problems in Section VII.

Notation: $\mathbb{R}$ denotes the set of real numbers. $\mathbb{N}$ denotes the set of integer numbers. $\mathbf{1}_n$ represents the n-dimensional vector of ones and I_n represents the identity matrix of order n. The symbol $|\cdot|$ denotes the cardinality of a set. For a vector $v = [v_1, \ldots, v_n]^\top \in \mathbb{R}^n$, the following notation is used:

$$\min(v) := \min_i v_i, \ \max(v) := \max_i v_i.$$

For a matrix $X \in \mathbb{R}^{m\times n}$, we use $\min(X)$ to denote the vector $[\min(X_{1*}), \cdots, \min(X_{m*})]^\top$ and use $\max(X)$ to denote the vector $[\max(X_{1*}), \cdots, \max(X_{m*})]^\top$, where $X_{1*}, \ldots, X_{m*}$ are the row vectors of X. For a real number $x \in \mathbb{R}$, $\lceil x \rceil$ denotes the smallest integer greater than or equal to x. Moreover, $[x]_-$ is defined as

$$[x]_- = \begin{cases} x & x \le 0, \\ 0 & x > 0. \end{cases}$$

If x is a vector, $[x]_-$ means that every entry of x takes the above function. For any $v_1, v_2 \in \mathbb{R}^{m\times n}$, we say $v_1 \succeq v_2$ if all the entries of $v_1 - v_2$ are nonnegative and $v_1 \preceq v_2$ if all the entries of $v_1 - v_2$ are nonpositive.

The *gradient* of a function $f(x)$ at a point $x \in \mathbb{R}^r$ is defined to be the column vector

$$\nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1}, \cdots, \frac{\partial f(x)}{\partial x_r}\right]^\top.$$

The *Hessian* of $f(x)$ at x is defined to be the symmetric $r \times r$ matrix having $\partial^2 f(x)/\partial x_i \partial x_j$ as the ij-th element:

$$\nabla^2 f(x) = \left[\frac{\partial^2 f(x)}{\partial x_i \partial x_j}\right]_{r\times r}.$$

II. Preliminaries and Problem Formulation

A. Preliminaries for Graphs

A *directed graph (digraph)* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a non-empty finite set $\mathcal{V}$ of elements called *nodes* and a finite set $\mathcal{E}$ of ordered pairs of nodes called *edges* . In $\mathcal{G}$, a node i is said to be *reachable* from a node j if there exists a path from j to i. Moreover, $\mathcal{G}$ is said to be *strongly connected* if every node is reachable from every other node.

A *time-varying digraph* $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$ represents a digraph whose edge set changes over time. For a time interval $[k_1, k_2]$, the *union digraph* is defined as $\mathcal{G}([k_1, k_2]) := (\mathcal{V}, \bigcup_{k\in[k_1,k_2]} \mathcal{E}(k))$. A time-varying digraph $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$ is said to be *jointly strongly connected* if there exists K such that for every k_0 the union digraph $\mathcal{G}([k_0, k_0 + K))$ is strongly connected. We call K the *period* of $\mathcal{G}(k)$.

For each node $i \in \mathcal{V}$, let $\mathcal{N}_i^+(k) := \{j \in \mathcal{V} : (j, i) \in \mathcal{E}(k)\}$ denote the set of its *in-neighbors* , and let $\mathcal{N}_i^-(k) := \{l \in \mathcal{V} : (i, l) \in \mathcal{E}(k)\}$ denote the set of its *out-neighbors* . Note that at any time k, $\mathcal{N}_i^+(k) \neq \mathcal{N}_i^-(k)$ generally. In addition, $i \notin \mathcal{N}_i^+(k)$ and $i \notin \mathcal{N}_i^-(k)$. Let $d_i^+(k) := |\mathcal{N}_i^+(k)|$ be its *in-degree* and let $d_i^-(k) := |\mathcal{N}_i^-(k)|$ be its *out-degree* .

B. Distributed Resource Allocation Problem

We consider a distributed resource allocation problem over a network of autonomous agents. The network is modelled as a time-varying digraph $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$ with node set $\mathcal{V} = \{1, 2, \ldots, n\}$ and edge set $\mathcal{E}(k) \subseteq \mathcal{V} \times \mathcal{V}$. By this model, each agent is only capable of receiving messages from its in-neighbors and transmitting messages to its out-neighbors. Moreover, the neighboring relationship also changes over time due to unpredictable packet losses or deliberate operations.

Each agent i of the network is associated with a local variable (resource) $x_i \in \mathbb{R}$ and a convex cost function $F_i : \mathbb{R} \to \mathbb{R}$. The local variable (resource) $x_i \in \mathbb{R}$ is subject to an inequality constraint $\underline{x}_i \le x_i \le \overline{x}_i$. Moreover, the sums of combinations of multiple variables (resources) across the network are fixed. To sum up, the optimal resource allocation problem is stated as follows.

$$\underset{x_1,\ldots,x_n}{\text{minimize}} \quad \sum_{i=1}^n F_i(x_i) \tag{1a}$$

subject to:

$$\underline{x}_i \le x_i \le \overline{x}_i, \forall i, \quad \text{(individual state constraints)} \tag{1b}$$

$$Cx = d, \quad \text{(collective equality constraints)} \tag{1c}$$

where $x = [x_1, \cdots, x_n]^\top$, $d = [d_1, \cdots, d_m]^\top$, and $C \in \mathbb{R}^{m\times n}$ with every column having only one entry being 1 and others being 0. We can think of d_i, $i = 1, \ldots, m$, as the total amount of a type of resources, that is only allowed to be allocated to a subset of agents.

In this paper, we are interested in distributed algorithms for solving this general problem (1), where each agent is only allowed to conduct local computation via received information from its in-neighbors. Thus, the local information structure imposed by the time-varying digraph should be considered as part of the problem formulation, which makes the problem extraordinarily challenging. We assume that the functions F_i are strongly convex and twice continuously differentiable with the second derivatives that are bounded below as in [3], [6].

Assumption 1: The function $F_i(x_i)$ is twice continuously differentiable in $\mathbb{R}$ and the second derivative is lower-bounded in the interval $[\underline{x}_i, \overline{x}_i]$, i.e.,

$$\frac{dF_i^2(x_i)}{dx_i^2} \ge l_i > 0, \forall \underline{x}_i \le x_i \le \overline{x}_i,$$

where l_i is a constant. ▲

Moreover, the following assumption is made for the time-varying digraph $\mathcal{G}(k)$.

Assumption 2: The time-varying digraph $\mathcal{G}(k)$ is jointly strongly connected with period K. ▲

Remark 1: If a solution exists and satisfies (1b) and (1c), we call it a feasible solution to the resource allocation problem (1). As each column of C has only one nonzero entry, it can be verified that there is a feasible solution for the resource allocation problem (1) if and only if

$$C\underline{x} \preceq d \preceq C\overline{x}$$

where $\underline{x} = [\underline{x}_1, \cdots, \underline{x}_n]^\top$ and $\overline{x} = [\overline{x}_1, \cdots, \overline{x}_n]^\top$. ▲

III. Nonnegative-Surplus Based Algorithm

In this section, we are going to develop a distributed algorithm to solve the resource allocation problem (1).

For each agent i, we define the incremental cost function

$$J_i(x_i) := \frac{dF_i(x_i)}{dx_i}. \tag{2}$$

Moreover, we denote

$$f(x) = \sum_{i=1}^n F_i(x_i),$$

$$h(x) = d - Cx, \text{ and}$$

$$g(x) = [x_1 - \overline{x}_1, \cdots, x_n - \overline{x}_n, \underline{x}_1 - x_1, \cdots, \underline{x}_n - x_n]^\top.$$

Let $X = \{x | h(x) = 0,\ g(x) \preceq 0\}$ be the set of all feasible solutions. The Lagrangian function is constructed as

$$L(x, \nu, \mu) = \sum_{i=1}^n F_i(x_i) - \nu^\top(Cx - d) + \sum_{i=1}^n \mu_i(x_i - \overline{x}_i) + \sum_{i=1}^n \mu_{n+i}(\underline{x}_i - x_i),$$

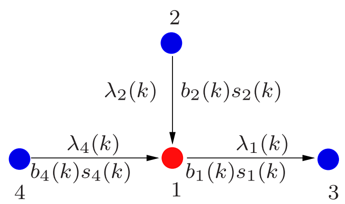


Fig. 1. Information exchanging in Algorithm 1.

where ν and $\mu = [\cdots, \mu_i, \cdots, \mu_{n+i}, \cdots]^\top$ are the Lagrange multipliers.

Now, we present the optimality condition for the resource allocation problem (1), which can be simply obtained from the Lagrangian equivalence.

Lemma 1: (Proposition 1.29, 1.30, Page 71, [32]) Under Assumption 1, x^* is the globally optimal solution to the resource allocation problem (1) and (ν^*, μ^*) are the optimal Lagrange multipliers if and only if

$$\begin{cases} J_i(x_i^*) = C_{*i}^\top \nu^* & \text{for } \underline{x}_i < x_i^* < \overline{x}_i, \\ J_i(x_i^*) \le C_{*i}^\top \nu^* & \text{for } x_i^* = \overline{x}_i, \\ J_i(x_i^*) \ge C_{*i}^\top \nu^* & \text{for } x_i^* = \underline{x}_i, \end{cases} \tag{3}$$

where C_{*i} denotes the i-th column of C. ▲

The condition (3) can also be written in the following equivalent form: For every i,

$$x_i^* = \phi_i(C_{*i}^\top \nu^*) \tag{4}$$

where

$$\phi_i(C_{*i}^\top \nu^*) := \begin{cases} \overline{x}_i & \text{if } C_{*i}^\top \nu^* > J_i(\overline{x}_i) \\ J_i^{-1}(C_{*i}^\top \nu^*) & \text{if } J_i(\underline{x}_i) \le \ C_{*i}^\top \nu^* \le J_i(\overline{x}_i) \\ \underline{x}_i & \text{if } C_{*i}^\top \nu^* < J_i(\underline{x}_i) \end{cases} \tag{5}$$

with $J_i^{-1}(\cdot)$ being the inverse function of $J_i(x_i)$. From (4), we know that if the optimal Lagrange multiplier ν^* is obtained, then the globally optimal solution x_i^* can be obtained in a distributed manner. Based on this observation, our main idea is to let each agent have its own copy of the Lagrange multiplier, say $\lambda_i \in \mathbb{R}^m$, and update λ_i such that all λ_i reach consensus at ν^*. Also, at each step, each agent calculates its estimate x_i about the optimal x_i^* according to the projection map $x_i = \phi_i(C_{*i}^\top \lambda_i)$, which confines the estimate x_i in the interval $[\underline{x}_i, \overline{x}_i]$. However, due to asymmetrical information flow in the time-varying digraph model and also due to the nonlinear projection map $x_i = \phi_i(C_{*i}^\top \lambda_i)$, the estimate $[x_1, \ldots, x_n]^\top$ at each step may not be a feasible solution to (1c) though it starts with a feasible solution of (1c). In order to overcome this challenge, a surplus variable $s_i \in \mathbb{R}^m$ is introduced for each agent i to temporarily store the resulting bias that will then be averaged with its neighbors such that it vanishes asymptotically. These ideas are summarized in the algorithm below.

The operator $[\cdot]_-$ for the item $\sum_{j\in\mathcal{N}_i^+(k)} a_i(k)(\lambda_j(k) - \lambda_i(k))$ in Algorithm 1 ensures non-negative s_i as shown in the following lemma. So we call Algorithm 1 a *nonnegative-surplus based algorithm* .

Lemma 2: For Algorithm 1, if $s_i(0) \succeq 0$ for all $i \in \mathcal{V}$, then $s_i(k) \succeq 0$ for all $i \in \mathcal{V}$ and $k \ge 0$. ▲

Algorithm 1: Nonnegative-Surplus Based Algorithm.

Initialization:

(1) Choose

$$x_i(0) \in [\underline{x}_i, \overline{x}_i] \text{ and } s_i(0) \succeq 0 \text{ for all } i \tag{6}$$

such that

$$Cx(0) + \sum_{i=1}^n s(0) = d; \tag{7}$$

(2) Choose $\lambda_i(0)$ such that

$$C_{*i}^\top \lambda_i(0) = J_i(x_i(0)). \tag{8}$$

Update:

$$\lambda_i(k+1) = \lambda_i(k) + \left[\sum_{j\in\mathcal{N}_i^+(k)} a_i(k)(\lambda_j(k) - \lambda_i(k))\right]_- + \epsilon_i(k)s_i(k), \tag{9a}$$

$$x_i(k+1) = \phi_i(C_{*i}^\top \lambda_i(k+1)), \tag{9b}$$

$$s_i(k+1) = b_i(k)s_i(k) + \sum_{j\in\mathcal{N}_i^+(k)} b_j(k)s_j(k) - C_{*i}(x_i(k+1) - x_i(k)), \tag{9c}$$

where the parameters in the algorithm are chosen as follows: $a_i(k) = \frac{1}{d_i^+(k)+1}, b_i(k) = \frac{1}{d_i^-(k)+1}$ and $\epsilon_i(k) = \widehat{c}_i b_i(k)$ with $\widehat{c}_i \in (0, l_i)$.

The proof of Lemma 2 is given in the appendix.

In the following, we briefly explain how Algorithm 1 is implemented in a distributed fashion based on information exchange among neighbors. For Algorithm 1, each agent i only needs to know its in-degree $d_i^+(k)$ and out-degree $d_i^-(k)$. In addition, each agent i receives $\lambda_j(k)$ and $b_j(k)s_j(k)$ from its in-neighbors, and transmits $\lambda_i(k)$ and $b_i(k)s_i(k)$ to its out-neighbors. See Fig. 1 for an example on what information being exchanged in the network.

Moreover, the design of the parameters used in Algorithm 1 does not require any global knowledge of the network, but local knowledge from itself and its neighbors.

Remark 2: Recall that in Algorithm 1,

$$a_i(k) = \frac{1}{d_i^+(k)+1},\ b_i(k) = \frac{1}{d_i^-(k)+1},\ \text{and } \epsilon_i(k) = \widehat{c}_i b_i(k)$$

with $\widehat{c}_i$ being a constant in $(0, l_i)$. So $a_i(k)$, $b_i(k)$, and $\epsilon_i(k)$ are uniformly bounded from below. That is, there exist $\varrho_a, \varrho_b, \varrho_c > 0$ such that

$$\varrho_a \le a_i(k) \le 1,\ \varrho_b \le b_i(k) \le 1,\ \text{and } \varrho_c \le \epsilon_i(k) < l_i.$$

Note that for any constant $\widehat{c}_i \in (0, l_i)$, there exists $\varrho_d > 0$ such that

$$\varrho_d \le (1 - \widehat{c}_i / l_i) b_i(k) < 1.$$

In the following, we denote $\varrho := \min\{\varrho_a, \varrho_b, \varrho_c, \varrho_d\}$. ▲

Remark 3: For a static digraph model, a similar idea using a surplus variable is considered in [3] to solve the resource allocation problem. However, it has two limitations. First, the algorithm in [3] may not converge when the digraph becomes time-varying as shown in our simulation section later. Second, a critical parameter requires global knowledge of the whole network in the design step; Otherwise, the convergence may not be guaranteed. Our algorithm, however, does not have such drawbacks. ▲

Remark 4: In different applications, there may or may not exist self-loops. We note, however, that our algorithm uses only relative information between agents, and as a result there is no change to the algorithm whether with or without self-loops. (Relative information of an agent with respect to itself is zero.) ▲

IV. Convergence Analysis

In this section, we show that the nonnegative-surplus based algorithm (Algorithm 1) converges and provides the globally optimal solution to the resource allocation problem (1).

Theorem 1: Suppose Assumptions 1 and 2 hold. Algorithm 1 converges to the globally optimal solution to the resource allocation problems (1). ▲

To prove Theorem 1, we present several technical lemmas, but the proofs of these technical lemmas are given in the appendix.

The first lemma shows that the collective equality constraints hold by counting into the surplus values when the agents run Algorithm 1.

Lemma 3: $Cx(k) + \sum_{i=1}^{n} s_i(k) = d$ for all k. ▲

The second lemma shows that every component of the surplus vector $s_i(k)$ is upper bounded under Algorithm 1.

Lemma 4: Suppose Assumption 2 holds. Then for all $i \in \mathcal{V}$ and $k \geq 0$, it holds that

$$s_i(k) \preceq \frac{\min(\lambda(k+\Delta+1)) - \min(\lambda(k+\Delta))}{\varrho^{\Delta+1}}, \tag{10}$$

where $\Delta = (n-1)K$. ▲

The following lemma presents several properties about $\lambda_i(k)$. In what follows, we denote $\lambda := [\lambda_1, \cdots, \lambda_n]$ and $s := [s_1, \cdots, s_n]$, which are $m \times n$ matrices.

Lemma 5: For Algorithm 1, the following holds:

1) Every component of $\min(\lambda(k))$ is non-decreasing with respect to k;
2) There exists a vector $\bar{\lambda} \in \mathbb{R}^m$ such that $\min(\lambda(k)) \preceq \bar{\lambda}$;
3) There exists a constant vector $\sigma = [\sigma_1, \cdots, \sigma_m]^\top \in \mathbb{R}^m$ such that $\lim_{k\to\infty} \min(\lambda(k)) = \sigma$. ▲

Now we are ready to present the main proof for Theorem 1

Proof of Theorem 1: From Lemma 4 and Lemma 5, it can be inferred that

$$\lim_{k\to\infty} s(k) = 0.$$

Next, we show that all λ_i converge to the same vector. Look at each component of s_i and λ_i and denote by s_{hi} and λ_{hi} the h-th component of s_i and λ_i, respectively. Moreover, denote $\lambda_{h*} = [\lambda_{h1}, \cdots, \lambda_{hn}]$ and denote $s_{h*} = [s_{h1}, \cdots, s_{hn}]$. Thus, it is equivalent to show that for every $h = 1, \ldots, m$, all λ_{hi} converge to the same value.

First, we show the limit of every $\lambda_{hi}(k)$ exists, that is, λ_{hi} is bounded and not oscillatory. For any $i \in \mathcal{V}$ and $k \geq k_0 \geq 0$, by (9a) we obtain

$$\begin{aligned} \lambda_{hi}(k) &\leq \lambda_{hi}(k_0) + \sum_{k'=k_0}^{k-1} \epsilon_i(k') s_{hi}(k') \\ &\leq \lambda_{hi}(k_0) + g_{hi}(k_0), \end{aligned} \tag{11}$$

where

$$g_{hi}(k) := \sum_{k'=k}^{\infty} \epsilon_i(k') s_{hi}(k').$$

The following proof is divided into three steps.

Step 1: To show that $g_{hi}(k)$ is bounded and $\lim_{k\to\infty} g_{hi}(k) = 0$ for every i.

By the fact that $s_{hi}(k) \geq 0$, we obtain $g_{hi}(k) \geq 0$. Then from Lemma 4, it follows that

$$\begin{aligned} g_{hi}(k) &= \sum_{k'=k}^{\infty} \epsilon_i(k') s_{hi}(k') \\ &\leq l_i \sum_{k'=k}^{\infty} \frac{\min(\lambda_{h*}(k'+\Delta+1)) - \min(\lambda_{h*}(k'+\Delta))}{\varrho^{\Delta+1}} \\ &= l_i \frac{\lim_{k'\to\infty} \min(\lambda_{h*}(k'+\Delta+1)) - \min(\lambda_{h*}(k+\Delta))}{\varrho^{\Delta+1}} \\ &= l_i \frac{\sigma_h - \min(\lambda_{h*}(k+\Delta))}{\varrho^{\Delta+1}}. \end{aligned}$$

Using Lemma 5, it is then clear that $g_{hi}(k)$ is bounded and $\lim_{k\to\infty} g_{hi}(k) = 0$.

Step 2: To show that $\lambda_{hi}(k)$ is bounded for any i.

Recall that

$$\min(\lambda_{h*}(k)) \leq \lambda_{hi}(k) \leq \lambda_{hi}(k_0) + g_{hi}(k_0).$$

Since $g_{hi}(k)$ is bounded and $\min(\lambda_{h*}(k))$ is non-decreasing, it can be obtained that $\lambda_{hi}(k)$ is bounded.

Step 3: To show that $\lambda_{hi}(k)$ is not oscillatory for any i.

Suppose on the contrary that there exists an agent $i \in \mathcal{V}$ such that $\lambda_{hi}(k)$ is oscillating. Then we can construct two subsequences $\{k'_0, k'_1, \ldots\}$ and $\{k''_0, k''_1, \ldots\}$ such that $k'_j > k''_j$ for any $j = 0, 1, \ldots$, and $\{\lambda_{hi}(k'_j)\}$ and $\{\lambda_{hi}(k''_j)\}$ have two different limits, i.e., $\lim_{j\to\infty} \lambda_{hi}(k'_j) = a$, $\lim_{j\to\infty} \lambda_{hi}(k''_j) = b$, and $a \neq b$. Without loss of generality, we assume $a > b$. Then from (11) we can get that

$$\lim_{j\to\infty} g_{hi}(k''_j) \geq \lim_{j\to\infty} (\lambda_{hi}(k'_j) - \lambda_{hi}(k''_j)) = a - b > 0,$$

a contradiction to $\lim_{k\to\infty} g_{hi}(k) = 0$. Thus, we conclude that $\lambda_{hi}(k)$ has a limit for any i.

Second, we show that all $\lambda_{hi}(k)$ converge to σ_h.

There are two possible cases.

1) $\lambda_{hi}(k)$ converge to the same value for all i;
2) $\lambda_{hi}(k)$ converge to different values for different i.

Consider case (i) and recall that $\lim_{k\to\infty} \min(\lambda_{h*}(k)) = \sigma_h$. It can be directly obtained that

$$\lim_{k\to\infty} \lambda_{h*}(k) = \sigma_h \mathbf{1}_n^\top.$$

Next we prove that case (ii) will not happen. We suppose on the contrary that $\lambda_{hi}(k)$'s converge to different values for different i. Again with the fact that $\lim_{k\to\infty}\min(\lambda_{h*}(k)) = \sigma_h$, there exists i such that

$$\lim_{k\to\infty}\lambda_{hi}(k) > \sigma_h.$$

Then we can relabel the nodes if necessary such that

$$\begin{cases} \lim_{k\to\infty}\lambda_{hi}(k) = \sigma_{h1},\ i = 1,\ldots,n_1 \\ \lim_{k\to\infty}\lambda_{hi}(k) = \sigma_{h2},\ i = n_1+1,\ldots,n_2 \\ \qquad\vdots \\ \lim_{k\to\infty}\lambda_{hi}(k) = \sigma_{hp},\ i = n_{p-1}+1,\ldots,n \end{cases}$$

and

$$\sigma_{h1} > \sigma_{h2} > \ldots > \sigma_{hp} = \sigma_h.$$

We choose

$$\varepsilon = \frac{1}{5}\varrho(\sigma_{h1} - \sigma_{h2})$$

and it is certain that there exists a k_1 such that for any $k > k_1$,

$$\begin{cases} \sigma_{h1} - \varepsilon < \lambda_{hi}(k) < \sigma_{h1} + \varepsilon,\ i = 1,\ldots,n_1 \\ \qquad\vdots \\ \sigma_{hp} - \varepsilon < \lambda_{hi}(k) < \sigma_{hp} + \varepsilon,\ i = n_{p-1}+1,\ldots,n \end{cases}$$

Since $\mathcal{G}(k)$ is jointly strongly connected with period K, for any interval $[k, k+K)$ $(k > k_1)$, there exists a $k' \in [k, k+K)$, $i \in \{1,\ldots,n_1\}$ and $i' \in \{n_1+1,\ldots,n\}$ such that $(i', i) \in \mathcal{E}(k')$. Then from (9a), we have

$$\begin{aligned} &\epsilon_i(k')s_{hi}(k') \\ &= \lambda_{hi}(k'+1) - \lambda_{hi}(k') - \left[\sum_{j\in\mathcal{N}_i^+} a_i(\lambda_{hj}(k') - \lambda_{hi}(k'))\right]_- \\ &> -4\varepsilon + \varrho(\sigma_{h1} - \sigma_{h2}) \\ &= \varepsilon > 0, \end{aligned}$$

a contradiction to $\lim_{k\to\infty} s_{hi}(k) = 0$.
To conclude,

$$\lim_{k\to\infty}\lambda_{h*}(k) = \sigma_h \mathbf{1}_n^\top.$$

Together with the fact h can be any value in $\{1,\cdots,m\}$, we obtain

$$\lim_{k\to\infty}\lambda(k) = \boldsymbol{\sigma}\mathbf{1}_n^\top.$$

Therefore, the conclusion follows. ■

V. Initialization and Feasibility Analysis

In this section, we come to discuss how to find an initialization for Algorithm 1 and also investigate on how to verify the existence of a solution to the resource allocation problem (1). To provide an initialization for Algorithm 1 is equivalent to find a feasible solution satisfying (1b) and (1c). In this section, we will provide a distributed algorithm for the initialization of Algorithm 1 as well as for the feasibility test to check whether the resource allocation problem (1) has a solution.

The following lemma states what we need.

Lemma 6: If $C\underline{x} \preceq d \preceq C\overline{x}$, then the optimal solution to the following optimization problem:

$$\underset{x_1,\ldots,x_n}{\text{minimize}} \quad \sum_{i=1}^n \bar{F}_i(x_i) \tag{12a}$$

subject to

$$Cx = d, \tag{12b}$$

where $\bar{F}_i(x_i) = \frac{1}{2}\left(\frac{x_i - \underline{x}_i}{\overline{x}_i - \underline{x}_i}\right)^2$, is a feasible solution satisfying (1b) and (1c). ▲

Proof: For the h-th row of C, without loss of generality, we assume that $C_{hi} = 1$ for $i \in \{h_1,\cdots,h_p\}$ and $C_{hi} = 0$ for $i \notin \{h_1,\cdots,h_p\}$.

By Lemma 1, we know that if x^* is the optimal solution to (12), then the following holds:

$$\frac{x^*_{h_1} - \underline{x}_{h_1}}{\overline{x}_{h_1} - \underline{x}_{h_1}} = \cdots = \frac{x^*_{h_p} - \underline{x}_{h_p}}{\overline{x}_{h_p} - \underline{x}_{h_p}} = \frac{d_h - \sum_{i=h_1}^{h_p}\underline{x}_i}{\sum_{i=h_1}^{h_p}\overline{x}_i - \sum_{i=h_1}^{h_p}\underline{x}_i}. \tag{13}$$

By the condition $C\underline{x} \preceq d \preceq C\overline{x}$ and also the assumption that every column of C has only one entry being 1, it follows that

$$0 \le \frac{d_h - \sum_{i=h_1}^{h_p}\underline{x}_i}{\sum_{i=h_1}^{h_p}\overline{x}_i - \sum_{i=h_1}^{h_p}\underline{x}_i} \le 1,$$

which implies

$$\overline{x}_i \le x_i^* \le \underline{x}_i,\ \forall i.$$

So it is a feasible solution satisfying (1b) and (1c). ■

Note that the optimization problem (12) is a special case of (1) without individual state constraints. Moreover, the cost function in (12a) is of particular form

$$\bar{F}_i(x_i) = \frac{1}{2}\left(\frac{x_i - \underline{x}_i}{\overline{x}_i - \underline{x}_i}\right)^2.$$

Thus, the optimization problem (12) can be solved in a distributed way by almost the same algorithm as Algorithm 1. But since there is no individual state constraint for the optimization problem (12), the initialization starts with a feasible solution only satisfying (1c) and the projection map (9b) in Algorithm 1 also changes. The following algorithm summarizes the idea.

For the initialization in Algorithm 2, leader election is one approach. That is, m tokens are passed in the network. If any agent j with nonzero c_{ij} (the (i,j)-th entry of C) gets token i, and also knows d_i (the i-th entry of d), then it chooses

$$x_j(0) = d_i$$

and stops passing token i. For those who do not get any token, they choose their initial states to be 0.

An alternative distributed approach is to use "integer consensus" [33]. Suppose that each agent has a unique ID, an integer number. For those who know d_i, running the integer consensus algorithm will let them reach the same integer. The agent whose ID is the same as the integer wins and sets its initial state to be d_i.

As Algorithm 2 is a special form of Algorithm 1, the convergence is also guaranteed by the same analysis as done in

Algorithm 2: Distributed Algorithm for Problem (12).

Initialization:

Choose $s_i(0) = 0$ and $x_i(0), i = 1, \ldots, n$, such that

$$Cx(0) = d.$$

Moreover, choose $\eta_i(0),\ i = 1, \ldots, n$, such that

$$C_{*i}^{\top}\eta_i(0) = \frac{d\bar{F}_i(x_i)}{dx_i}|_{x_i = x_i(0)} := \frac{x_i(0) - \underline{x}_i}{\overline{x}_i - \underline{x}_i}.$$

Update:

$$\eta_i(k+1) = \eta_i(k) + \left[\sum_{j \in \mathcal{N}_i^+(k)} a_i(k)(\eta_j(k) - \eta_i(k))\right]_- + \epsilon_i(k)s_i(k), \quad (14a)$$

$$x_i(k+1) = \psi_i(C_{*i}^{\top}\eta_i(k+1)) \coloneqq (\overline{x}_i - \underline{x}_i)C_{*i}^{\top}\eta_i(k+1) + \underline{x}_i, \quad (14b)$$

$$s_i(k+1) = b_i(k)s_i(k) + \sum_{j \in \mathcal{N}_i^+(k)} b_j(k)s_j(k) - C_{*i}(x_i(k+1) - x_i(k)), \quad (14c)$$

where the parameters in the algorithm are chosen as follows: $a_i(k) = \frac{1}{d_i^+(k)+1}, b_i(k) = \frac{1}{d_i^-(k)+1}$ and $\epsilon_i(k) = c_i \bar{l}_i b_i(k)$ with $c_i \in (0, 1)$ and $\bar{l}_i = \frac{1}{\overline{x}_i - \underline{x}_i}$.

TABLE I
SIMULATION PARAMETERS

Agents Type	A	B	C
State Constraints	$[0.5, 2]$	$[-0.5, 1]$	$[-1, 1]$
Cost Function	x_i^3	$x_i^3 + 3x_i^2$	x_i^2
Incremental Cost	$3x_i^2$	$3x_i^2 + 6x_i$	$2x_i$

Section IV. That is, all $\eta_i(k)$ in Algorithm 2 converge to the optimal Lagrange multiplier. We denoted $\eta^* = \lim_{k\to\infty} \eta_i(k)$ for all i. The following lemma tells whether the resource allocation problem (1) has a solution.

Lemma 7: If $0 \le C_{*i}^{\top}\eta^* \le 1$ for all i, then the resource allocation problem (1) has a feasible solution; Otherwise, it has no feasible solution.

Proof: By Algorithm 2, we obtain that $C_{*i}^{\top}\eta^* = \frac{x_i^* - \underline{x}_i}{\overline{x}_i - \underline{x}_i}$. Thus, if $0 \le C_{*i}^{\top}\eta^* \le 1$ for all i, it follows from the formula (13) in the proof of Lemma 6 that

$$(\forall h \in \{1, \cdots, m\})\ C_{h*}\underline{x}_i \le d_h \le C_{h*}\overline{x}_i,$$

which can be rewritten as

$$C\underline{x} \preceq d \preceq C\overline{x}.$$

As pointed out in Remark 1, the resource allocation problem (1) has a feasible solution if and only if

$$C\underline{x} \preceq d \preceq C\overline{x}.$$

Therefore, the resource allocation problem (1) has a feasible solution.

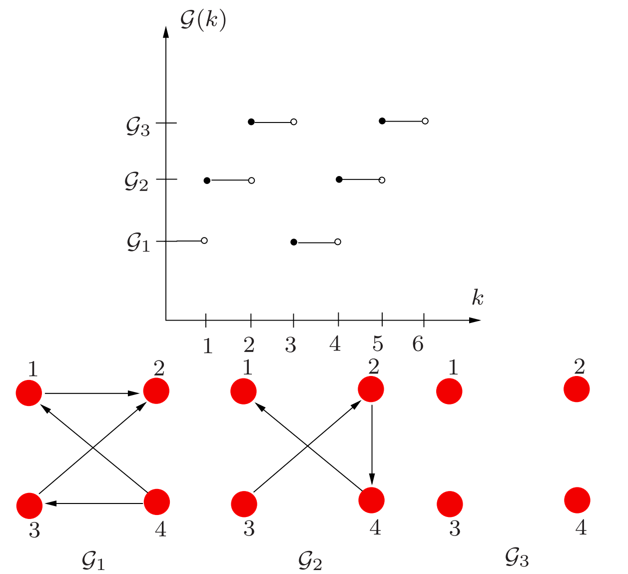


Fig. 2. A time-varying digraph $\mathcal{G}(k)$ that switches among three different topologies $\mathcal{G}_1$, $\mathcal{G}_2$ and $\mathcal{G}_3$, i.e., $(\forall k \ge 0)\mathcal{G}(3k) = \mathcal{G}_1$, $\mathcal{G}(3k+1) = \mathcal{G}_2$ and $\mathcal{G}(3k+2) = \mathcal{G}_3$.

TABLE II
INITIAL CONDITIONS

Variable	$i = 1$	$i = 2$	$i = 3$	$i = 4$
$x(0)$	0.5	0.5	-0.5	-1
$s(0)$	0	0	0	6.5
$\lambda(0)$	0.75	0.75	24	-2

On the contrary, if there exists i such that $C_{*i}^{\top}\eta^* > 1$ or $C_{*i}^{\top}\eta^* < 0$, then from (13) we know that there exists a row h such that

$$d_h > C_{h*}\overline{x}_i \text{ or } d_h < C_{h*}\underline{x}_i.$$

Therefore, the resource allocation problem (1) has no feasible solution. ■

Remark 5: From the above analysis, we can know that if eqs. (1b)-(1c) has a unique solution, then this unique solution must take its value on the boundary of the interval $[\underline{x}_i, \overline{x}_i]$ for all i and Algorithm 2 takes infinite number of steps to find this unique feasible solution. But for this case, this unique solution is also the globally optimal solution to the resource allocation problem (1) and there is no need of running Algorithm 1. For other cases, if there is more than one feasible solution satisfying eqs. (1b)-(1c), then the optimal solution to the optimization problem (12) must take its value in the interior of $[\underline{x}_i, \overline{x}_i]$ for all i, which means that at some step k^*, all $C_{*i}^{\top}\eta_i(k^*)$ must be strictly inside the interval $(0, 1)$ for all i. Therefore, although it is not possible to know the explicit iteration number, a practical way can be adopted by each agent to stop Algorithm 2 when a valid initialization is ready to pick. For example, a very small positive parameter ϑ can be used to indicate whether Algorithm 2 converges close enough to the optimal solution to the problem (12) by checking whether $\|s_i(k^*)\| \le \vartheta$ for all i at step k^* and remains inside for sufficient long time. If so, $x_i(k^*)$ and $s_i(k^*)$ obtained from Algorithm 2 can be used as the initialization for Algorithm 1.

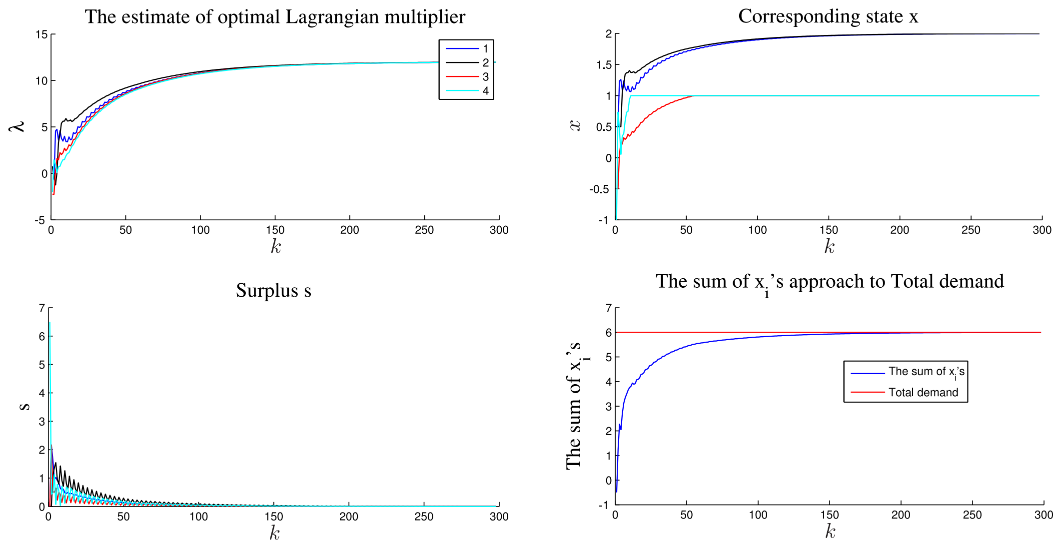


Fig. 3. The simulation trajectories of the estimated Lagrange multiplier, individual state, surplus, and sum of x_i's.

VI. Simulation Examples

In this section, we provide simulation examples to illustrate our proposed algorithms. Different from flow control with interrelated communication graph and collective equality constraints, these examples below imply that the communication graph and the collective equality constraints are separate in the resource allocation problem we study.

A. A Four-Agents Network Example

First, we consider a network with four agents, labeled as $1, 2, 3$, and 4. We summarize the state constraints and cost functions in Table I and the four agents are chosen from three types described in Table I. Note that the cost functions are strictly convex. Agent 1 and 2 are of Type A, agent 3 is of Type B, and agent 4 is of Type C. As an illustrative example, we consider a time-varying digraph $\mathcal{G}(k)$ switching among three different topologies as shown in Fig. 2.

We carry out a simulation with the following single collective equality constraint

$$\sum_{i=1}^{4} x_i = 6.$$

We set the initial conditions as in Table II.

The simulation results of Algorithm 1 are shown in Fig. 3. It can be checked that the optimal solution is: $\lambda^* = 11.98, x_1^* = 2, x_2^* = 2, x_3^* = 1, x_4^* = 1$. From the simulation results, we can see that by running Algorithm 1, all λ_i's converge to the optimal Lagrange multiplier λ^* as shown in the top-left sub-figure of Fig. 3. Each agent's state x_i converges to the optimal value x_i^* as shown in the top-right sub-figure. The bottom-left sub-figure shows that all the surplus values s_i's converge to 0 as desired. The sum of all x_i's is shown in the bottom-right, which converges to the constant in the collective equality constraint. For this simulation example, isolated agents exist when the topology becomes $\mathcal{G}_3$, during which $\lambda_i(k)$ are non-decreasing.

We also plot the difference between $x_i(k)$ and $J_i^{-1}(\lambda_i)$ in Fig. 4, from which we can see that the individual state constraint takes effect for those steps, during which the difference are nonzero.

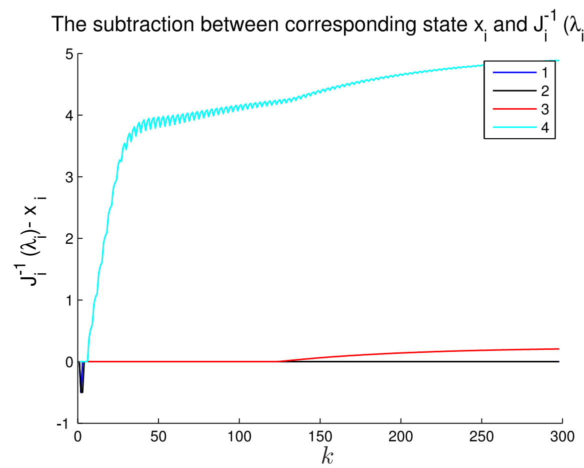


Fig. 4. The difference between $x_i(k)$ and $J_i^{-1}(\lambda_i(k))$.

Recall that $\widehat{c}_i$ can be written as $\widehat{c}_i = c_i l_i$ with $c_i \in (0, 1)$. Thus, to observe how the choice of $\widehat{c}_i$ affects the convergence rate of Algorithm 1, simulations are carried out for this example with different c_i. The simulation result plotted in Fig. 5 describes the relation between the number of iterations and the choice of c_i chosen from 0.01 to 0.99. It can be observed that the larger c_i is, the faster Algorithm 1 converges, which coincides with the intuition behind.

B. Simulations for Large-Scale Networks

In this subsection, we present several simulations to show the performance of our proposed algorithms for large-scale networks and compare with other existing algorithms.

In the first simulation, we consider a network of $n = 200$ agents. We randomly assign $\frac{(n-1)^2}{4}$ directed edges at each step. The cost function associated with each agent i takes the quadratic form $F_i(x_i) = \frac{(x_i - \alpha_i)^2}{2\beta_i} + \gamma_i$, where α_i, β_i, and γ_i

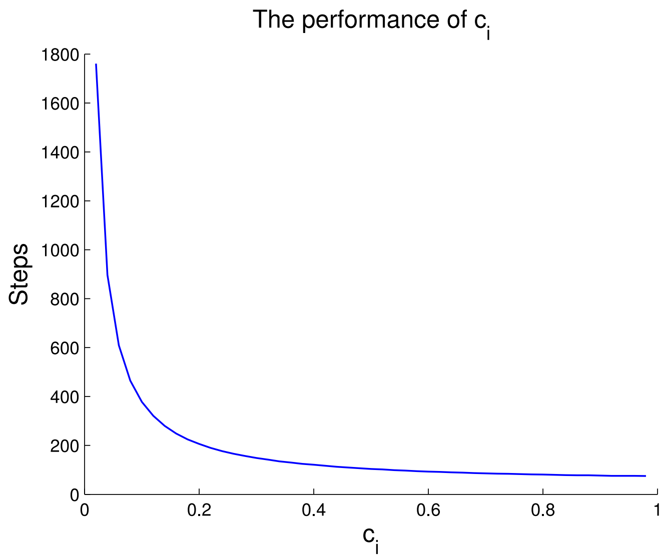


Fig. 5. The number of iterations (y-axis) for different c_i until $\|x(k) - x^*\| < 0.05$.

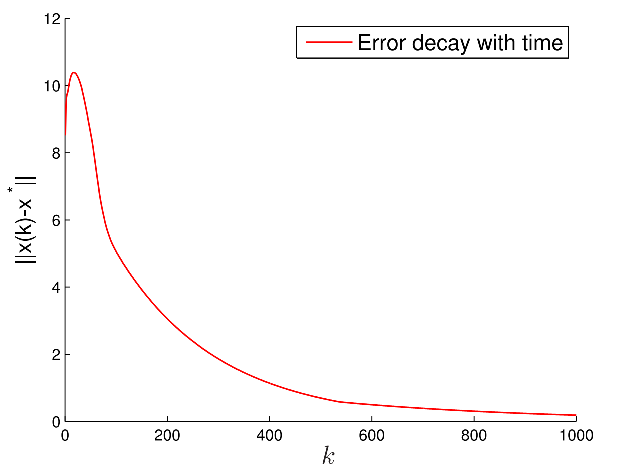


Fig. 6. The error norm $\|x(k) - x^*\|$ (y-axis) with respect to the iteration step (x-axis) for a network of 200 agents.

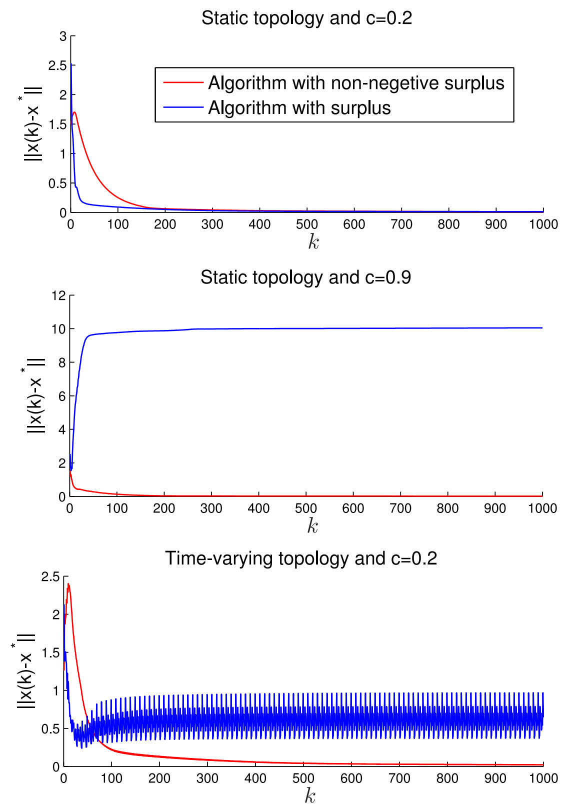


Fig. 7. Comparison of our algorithm with non-negative surplus and the algorithm with surplus in [3] for a network of 50 agents.

are randomly selected from $(0, 1)$ for each agent i. The individual state constraint is supposed to be the same for all the agents, namely, all constrained in $[-1, 1]$. Moreover, the collective equality constraint takes the vector of all 1's as C and $d = 10$. Algorithm 1 is run for this example with $c_i = 0.5$ for all i. The simulation result is shown in Fig. 6, which plots the average error $\|x(k) - x^*\|$ of 50 executions with respect to the iteration step. This simulation demonstrates that our proposed algorithm works well for large-scale networks.

In the second simulation, we consider a network of $n = 50$ agents. The cost functions and the state constraints adopt the same setup as in the first simulation. Three scenarios are simulated: (a) Consider a static digraph with $\frac{(n-1)^2}{2}$ directed edges randomly assigned for the network and use the parameter $c_i = 0.2$ for all i; (b) Consider a static digraph with $\frac{(n-1)^2}{2}$ directed edges randomly assigned for the network and use the parameter $c_i = 0.9$ for all i; (c) Consider a time-varying digraph with $\frac{(n-1)^2}{4}$ directed edges randomly assigned at each step for the network and use the parameter $c_i = 0.2$ for all i. For each scenario, both our algorithm (Algorithm 1) and the algorithm from [3] are run, where the parameter ϵ in the algorithm in [3] uses a value converted from the parameter c_i in our algorithm. The simulation results are plotted in Fig. 7, where the red curves are the error norm resulted from our algorithm and the blue curves are the error norm resulted from the algorithm in [3]. From the simulations, we can see that our algorithm converges to the optimal solution in all these three scenarios. The algorithm in [3] converges to the optimal solution for scenario a), which has a small parameter c_i, but it does not converge to the optimal solution for a large parameter c_i. Moreover, the error norm becomes oscillating and does not converge to zero when the digraph becomes time-varying.

At last, we show the scaling law of the period K and its influence on the convergence rate. Again, a network of $n = 50$ agents is considered. When counting the number of iterations, the termination condition $\|x(k) - x^*\| < 0.05$ is used. The plot in Fig. 8 is the averaged counting of 50 executions with respect to the period K from 1 to 10. The simulation shows that the number of iterations is nearly of linear growth with respect to the period K of a jointly strongly connected digraph.

C. Simulations for the IEEE 39-Bus system

In this subsection, we present a simulation for the IEEE 39-bus system with 10 generation units and 18 demand-side devices (Fig. 1 in [1]). The parameters are adopted from Table I in [2]. In smart grid, demands can be divided into two types:

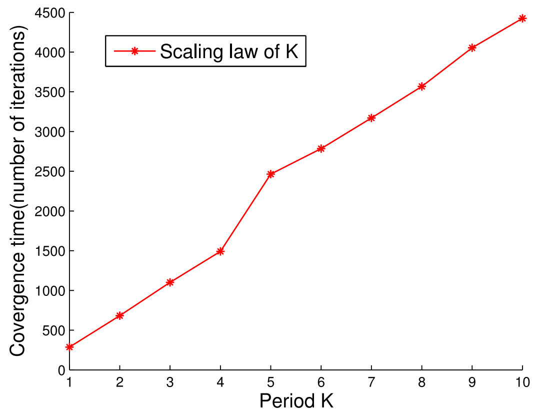


Fig. 8. The scaling law of K and its influence on the convergence rate.

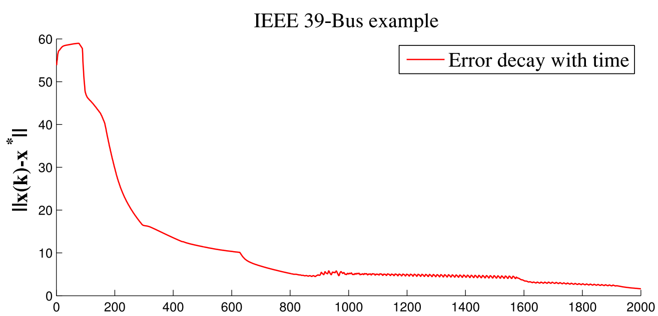


Fig. 9. The error norm $\|x(k) - x^*\|$ (y-axis) with respect to the iteration step (x-axis) for the IEEE 39-Bus system.

responsive demands and traditional demands [34]. The responsive demands have their own cost functions and state constraints, while the traditional demands are commonly fixed. In this example, we assume that the 18 demand-side devices are responsive demands and there exist extra traditional demands connecting to this system. That is, C is a $2 \times n$ matrix. Moreover, the estimates about the total generation, the total amount of traditional demands, and the total amount of responsive demands are $d_1 = 100, d_3 = 10$, and $d_2 = d_1 - d_3 = 90$, respectively. Thus, set $C_{1i} = 1, C_{2i} = 0$ for $i \leq 10$ and $C_{1i} = 0, C_{2i} = 1$ for $i > 10$, and set $x_i(0) = 0, s_{1i}(0) = 100, s_{2i}(0) = 90$ for $i = 1$ and $x_i(0) = 0, s_{1i}(0) = 0, s_{2i}(0) = 0$, otherwise . The simulation result is plotted in Fig. 9, which shows the convergence towards the optimal resource allocation solution.

VII. Conclusion

The distributed resource allocation problem has received increasing attention in recent years. However, most existing works assume either time-invariant or balanced communication topologies. This paper, however, removes the strong assumption on time-invariant or balanced topologies and develops a fully distributed algorithm for solving the resource allocation problem over arbitrary strongly connected digraphs. We introduce a nonnegative surplus to make the collective state constraints asymptotically satisfied and propose distributed iteration rules to steer the states to the optimal solution. The parameters used in the iterations do not require global knowledge, yet global convergence is assured.

Intuitively, a signed surplus scheme may lead to better convergence rate than our proposed nonnegative surplus scheme. However, it is still challenging on how to ensure global convergence by using only local knowledge for the design of the algorithm parameters. Further work will be continued to explore the inherent mechanisms for global convergence of distributed algorithms based on signed surplus variables. Besides, Assumption 1 indicates that the cost functions are strictly convex. Relaxation of this assumption is a further step towards general non-convex optimization problems. Moreover, by exploring further the inherent mechanism for using the surplus, it may become possible to solve the problem of allocating multiple resources that are coupled together. The main idea relies on how to update the surplus vector to ensure the convergence towards zero by projecting each individual's estimate about its own optimal amount of resources to the feasible solution space.

Appendix

The function $\phi_i(C_{*i}^\top \lambda_i)$ defined in (5) holds the following property, which is useful in the following proofs.

Lemma A.1: For each $i \in \mathcal{V}$, if $a \geq b$, then

$$0 \leq \phi_i(a) - \phi_i(b) \leq \frac{1}{l_i}(a - b),$$

where l_i is the constant in Assumption 1. ▲

Proof: By Assumption 1, we know that $J_i(x_i)$ is an increasing function, so is $J_i^{-1}(y_i)$. Thus, it follows from $d^2 F_i/dx_i^2 \geq l_i$ in Assumption 1 that $0 < dJ_i^{-1}(y_i)/dy_i \leq 1/l_i$. For any $a \geq b$, from (5) it is known that $\phi_i(a) - \phi_i(b) \geq 0$. On the other hand, by $0 < dJ_i^{-1}(y_i)/dy_i \leq \frac{1}{l_i}$ it can be inferred that

$$\phi_i(a) - \phi_i(b) \leq \frac{1}{l_i}(a - b). \quad \blacksquare$$

Proof of Lemma 2: By (9a) it follows that

$$\lambda_i(k+1) - \lambda_i(k) \preceq \epsilon_i(k) s_i(k).$$

Then if $C_{*i}^\top \lambda_i(k+1) \geq C_{*i}^\top \lambda_i(k)$, it follows from Lemma A.1 that

$$\begin{aligned} x_i(k+1) - x_i(k) &= \phi_i(C_{*i}^\top \lambda_i(k+1)) - \phi_i(C_{*i}^\top \lambda_i(k)) \\ &\leq \frac{1}{l_i}(C_{*i}^\top \lambda_i(k+1) - C_{*i}^\top \lambda_i(k)) \leq \frac{\epsilon_i(k) C_{*i}^\top s_i(k)}{l_i}. \end{aligned}$$

Moreover, notice there exist and only exist one positive value 1 for each column of C, and other elements are 0; without loss of generality, assume $C_{h_0 i} = 1$. Then, together with (9c), we have

$$\begin{aligned} & s_i(k+1) \succeq b_i(k) s_i(k) + \sum_{j \in \mathcal{N}_i^+(k)} b_j(k) s_j(k) - C_{*i} \frac{\epsilon_i(k)}{l_i} s_{h_0 j}(k) \\ & s_{h_0 i}(k+1) \geq (1 - c_i) b_i(k) s_{h_0 i}(k) + \sum_{j \in \mathcal{N}_i^+(k)} b_j(k) s_{h_0 j}(k) \\ \Rightarrow \quad & s_{hi}(k+1) \geq b_i(k) s_{hi}(k) + \sum_{j \in \mathcal{N}_i^+(k)} b_j(k) s_{hj}(k), h \neq h_0 \end{aligned} \tag{15}$$

On the other hand, if $C_{*i}^\top \lambda_i(k+1) < C_{*i}^\top \lambda_i(k)$, it can be obtained straightforward from (9c) that

$$s_i(k+1) \succeq b_i(k)s_i(k) + \sum_{j\in\mathcal{N}_i^+(k)} b_j(k)s_j(k). \tag{16}$$

Both $(1-c_i)b_i(k)$ and $b_j(k)$'s are positive. Hence, the conclusion that if $s_i(0) \succeq 0$ for all $i \in \mathcal{V}$, then

$$s_i(k) \succeq 0 \text{ for all } i \in \mathcal{V} \text{ and } k \geq 0.$$ ■

Proof of Lemma 3: By (9c), we obtain that

$$\begin{aligned}
&Cx(k+1) + \sum_{i=1}^n s_i(k+1)\\
&= \sum_{i=1}^n (s_i(k+1) + C_{*i}x_i(k+1))\\
&= \sum_{i=1}^n b_i(k)s_i(k) + \sum_{i=1}^n \sum_{j\in\mathcal{N}_i^+(k)} b_j(k)s_j(k) + \sum_{i=1}^n C_{*i}x(k)\\
&= \sum_{i=1}^n b_i(k)s_i(k) + \sum_{i=1}^n \sum_{j\in\mathcal{N}_i^-(k)} b_i(k)s_i(k) + Cx(k)\\
&= Cx(k) + \sum_{i=1}^n s_i(k).
\end{aligned}$$

Thus, the conclusion follows. ■

Proof of Lemma 4: For the time-varying digraph $\mathcal{G}(k)$, we define the *binary adjacency matrix* $A(k) = [e_{ij}(k)]$, where $e_{ij}(k) = 1$ if edge (j,i) exists at time k, and $e_{ij}(k) = 0$ otherwise. Moreover, recall that $s = [s_1, \cdots, s_n]$. Then it can be concluded from (15) in the proof of Lemma 2 that

$$s^\top(k+1) \succeq \varrho\,[I + A(k)]\,s^\top(k), \tag{17}$$

By repeatedly using (17), we know that for any $k' > k \geq 0$,

$$\begin{aligned}
s^\top(k') &\succeq \varrho^{k'-k}\,[I + A(k'-1)]\,[I + A(k'-2)]\\
&\quad \cdots [I + A(k)]\,s^\top(k)\\
&\succeq \varrho^{k'-k}\,[I + A(k) + \cdots + A(k'-1)]\,s^\top(k).
\end{aligned} \tag{18}$$

It then follows from (18) that

$$s_i(k') \succeq \varrho^{k'-k} s_i(k) \text{ for any } i. \tag{19}$$

For any $k \geq 0$, we consider the sequence of graphs $\mathcal{G}(k)$, $\mathcal{G}(k+1), \ldots, \mathcal{G}(k+\Delta)$ where $\Delta = (n-1)K$. Denote

$$\mathcal{G}_m \coloneqq \mathcal{G}([k+(m-1)K, k+mK-1]) \text{ for } m = 1, \ldots, n-1.$$

In addition, we denote

$$\widehat{A}_m := A(k+(m-1)K) + \cdots + A(k+mK-1),$$

which is the adjacency matrix of $\mathcal{G}_m$. By Assumption 2 that $\mathcal{G}(k)$ is jointly strongly connected, we know that $\mathcal{G}_1, \ldots, \mathcal{G}_{n-1}$ are all strongly connected with the same node set. Then it is clear that for any two nodes i and j, there exists a path $j \to l_{m-1} \to \cdots l_1 \to i$ of length $m \in \{1, \ldots, n-1\}$, for which edge $(j, l_{m-1}) \in \mathcal{G}_{(n-m)}$, …, edge $(l_2, l_1) \in \mathcal{G}_{(n-2)}$, and final edge $(l_1, i) \in \mathcal{G}_{(n-1)}$. It means by graph theory that the (i,j)-th entry of the matrix product

$$\Theta = [I + \widehat{A}_{(n-1)}][I + \widehat{A}_{(n-2)}] \cdots [I + \widehat{A}_{(n-m)}]$$

is positive. Then it follows from the inequality

$$s^\top(k+\Delta) \succeq \varrho^{mK}\Theta s^\top(k+(n-m-1)K)$$

that

$$s_i(k+\Delta) \succeq \varrho^{mK} s_j(k+(n-m-1)K). \tag{20}$$

Combining (19) and (20) leads to

$$s_i(k+\Delta) \succeq \varrho^{\Delta} s_j(k) \text{ for any } i \text{ and } j. \tag{21}$$

Looking at each component of s_i and λ_i. Denote s_{hi} and λ_{hi} as the h-th component of s_i and λ_i, $h \in \{1, \cdots, m\}$. Denote $\lambda_{h*} = [\lambda_{h1}, \cdots, \lambda_{hn}]$, $s_{h*} = [s_{h1}, \cdots, s_{hn}]$. For any $k \geq 0$, let $i_* \in \mathcal{V}$ be the node such that

$$\lambda_{hi_*}(k+\Delta+1) = \min(\lambda_{h*}(k+\Delta+1)). \tag{22}$$

Then by (21), it follows that

$$s_{hi}(k) \leq \frac{s_{hi_*}(k+\Delta)}{\varrho^{\Delta}}, \ \forall i \in \mathcal{V}. \tag{23}$$

Moreover, since $\lambda_i(k) + [\sum_{j\in\mathcal{N}_i^+(k)} a_i(k)(\lambda_j(k) - \lambda_i(k))]_-$ is a convex combination of $\lambda_i(k)$ and $\lambda_j(k) (j \in \mathcal{N}_i^+(k))$, we get

$$\lambda_{hi}(k) + \left[\sum_{j\in\mathcal{N}_i^+(k)} a_i(k)(\lambda_{hj}(k) - \lambda_{hi}(k))\right]_- \geq \min_{j\in\{i\}\bigcup\mathcal{N}_i^+(k)} \lambda_{hj}(k).$$

So by (9a), for any $i \in \mathcal{V}$ and $k \geq 0$, the following inequality holds:

$$\begin{aligned}
\lambda_{hi}(k+1) &\geq \min_{j\in\{i\}\bigcup\mathcal{N}_i^+(k)} \lambda_{hj}(k) + \epsilon_i(k)s_{hi}(k)\\
&\geq \min(\lambda_{h*}(k)) + \epsilon_i(k)s_{hi}(k)\\
&\geq \min(\lambda_{h*}(k)) + \varrho s_{hi}(k).
\end{aligned} \tag{24}$$

Recalling (22) and replacing i by i_* and k by $k+\Delta$ in (24), we obtain

$$s_{hi_*}(k+\Delta) \leq \frac{\min(\lambda_{h*}(k+\Delta+1)) - \min(\lambda_{h*}(k+\Delta))}{\varrho} \tag{25}$$

Plugging (25) into (23),

$$s_{hi}(k) \leq \frac{\min(\lambda_{h*}(k+\Delta+1)) - \min(\lambda_{h*}(k+\Delta))}{\varrho^{\Delta+1}}$$

Together with the fact h can be any value in $\{1, \cdots, m\}$, the conclusion is reached. ■

Proof of Lemma 5:

1) By Lemma 2 and (24) it can be obtained that

$$\lambda_i(k+1) \succeq \min(\lambda(k)),$$

which implies that every entry of $\min(\lambda(k))$ is non-decreasing with respect to k.

2) Consider any $h \in \{1, \ldots, m\}$. We let $\overline{\lambda}_h$ be the h-th component of $\overline{\lambda}$ and let λ_{h*} be the h-th row vector of λ. To prove 2), suppose on the contrary that for some h and for any $\overline{\lambda}_h$, there exists k such that

$$\min(\lambda_{h*}(k)) > \overline{\lambda}_h.$$

Without loss of generality, we assume that the h-th row vector of C is $C_{h*} = [\mathbf{1}_{n_1}^\top \ \mathbf{0}_{n-n_1}^\top], n_1 \leq n$.

Now we choose a particular $\overline{\lambda} = \eta$ such that

$$C_{*i}^\top \eta > J_i(\overline{x}_i), \ \forall i \in [1, \cdots, n_1].$$

Then, together with the component-wise non-decreasing property of $\min(\lambda(k))$, it follows that there exists a k_0 such that for all $k \geq k_0$,

$$C_{*i}^\top \lambda_i(k) > C_{*i}^\top \eta > J_i(\overline{x}_i), \ \forall i \in [1, \cdots, n_1].$$

Thus, by (9b), we have

$$x_i(k) = \overline{x}_i, \ \forall i \in [1, \cdots, n_1] \text{ and } \forall k \geq k_0.$$

Recall from Remark 1 that $d_h \leq C_{h*}\overline{x}$, where d_h is the h-th component of d. So from the fact that

$$C_{h*}x(k) + \sum_{i=1}^n s_{hi}(k) = d_h,$$

we attain

$$\sum_{i=1}^n s_{hi}(k) \leq 0, \ \forall k \geq k_0.$$

It follows from Lemma 2 that

$$\sum_{i=1}^n s_{hi}(k) \geq 0.$$

So the only possible case is that

$$\sum_{i=1}^n s_{hi}(k) = 0, \ \forall k \geq k_0. \tag{26}$$

We will show in the following that this is also not possible.

If (26) holds, then by the nonnegative properties of s_i, it follows that

$$s_{hi}(k) = 0, \ \forall i \in \mathcal{V} \text{ and } \forall k \geq k_0.$$

Thus, from (9a), we know for all i and for all $k \geq k_0$,

$$\lambda_{hi}(k+1) = \lambda_{hi}(k) + \left[\sum_{j \in \mathcal{N}_i^+(k)} a_i(k)(\lambda_{hj}(k) - \lambda_{hi}(k)) \right]_-,$$

where λ_{hi} is the (h,i)-th entry of λ. This implies $\lambda_{hi}(k) \leq \lambda_{hi}(k_0), \ \forall k \geq k_0$, a contradiction to the assumption that $\min(\lambda_{h*}(k))$ has no upper bound.

3) From 2), there exists a vector $\overline{\lambda} \in \mathbb{R}^m$ such that $\min(\lambda(k)) \preceq \overline{\lambda}$. We let $\sigma := \sup(\min(\lambda))$. Then according to the component-wise non-decreasing property in 1), it follows that

$$\lim_{k \to \infty} \min(\lambda(k)) = \sigma. \qquad \blacksquare$$

References

[1] C. Zhao, J. He, P. Cheng, and J. Chen, "Consensus-based energy management in smart grid with transmission losses and directed communication," *IEEE Trans. Smart Grid*, vol. PP, no. 99, pp. 1–13, doi: 10.1109/TSG.2015.2513772, preprint.

[2] N. Rahbari-Asr, U. Ojha, Z. Zhang, and M. Chow, "Incremental welfare consensus algorithm for cooperative distributed generation/demand response in smart grid," *IEEE Trans. Smart Grid*, vol. 5, no. 6, pp. 2836–2845, Nov. 2014.

[3] S. Yang, S. Tan, and J. Xu, "Consensus based approach for economic dispatch problem in a smart grid," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4416–4426, Nov. 2013.

[4] K. J. Morrisse, G. F. Solimini, and U. A. Khan, "Distributed control schemes for wind-farm power regulation," in *Proc. North Amer. Power Symp.*, Champaign, IL, USA, 2012, pp. 1–6.

[5] A. J. Wood and B. F. Wollenberg. *Power Generation, Operation, and Control*. Hoboken, NJ, USA: Wiley, 2012.

[6] L. Xiao and S. Boyd, "Optimal scaling of a gradient method for distributed resource allocation," *J. Optim. Theory Appl.*, vol. 129, no. 3, pp. 469–488, 2006.

[7] Z. Zhang and M. Chow, "Convergence analysis of the incremental cost consensus algorithm under different communication network topologies in a smart grid," *IEEE Trans. Power Syst.*, vol. 27, no. 4, pp. 1761–1768, Nov. 2012.

[8] S. Kar and G. Hug, "Distributed robust economic dispatch in power systems: A consensus+ innovations approach," in *Proc. Power Energy Soc. Gen. Meeting*, San Diego, CA, USA, 2012, pp. 1–8.

[9] V. Loia and A. Vaccaro, "Decentralized economic dispatch in smart grids by self-organizing dynamic agents," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 44, no. 4, pp. 397–408, Apr. 2014.

[10] H. Lim, V. Lam, M. Foo, and Y. Zeng, "An adaptive distributed resource allocation scheme for sensor networks," in *Proc. 2nd Int. Conf. Mobile Ad-hoc Sensor Netw.*, Hong Kong, 2006, pp. 770–781.

[11] V. C. Gungor *et al.*, "Smart grid technologies: Communication technologies and standards," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 151–164, Nov. 2011.

[12] E. Kranakis, D. Krizanc, and O. Morales, *Maintaining Connectivity in Sensor Networks Using Directional Antennae*. Berlin, Germany: Springer, 2010.

[13] A. D. Dominguez-Garcia, S. T. Cady, and C. N. Hadjicostis, "Decentralized optimal dispatch of distributed energy resources," in *Proc. 51st IEEE Conf. Decis. Control*, Maui, HI, USA, 2012, pp. 3688–3693.

[14] H. Xing, Y. Mou, M. Fu, and Z. Lin, "Distributed bisection method for economic power dispatch in smart grid," *IEEE Trans. Power Syst.*, vol. 30, no. 6, pp. 3024–3035, Nov. 2015.

[15] Y. C. Ho, L. D. Servi, and R. Suri, "A class of center-free resource allocation algorithms," *Large Scale Syst.*, vol. 1, pp. 51–62, 1980.

[16] S. H. Low and D. E. Lapsley, "Optimization flow control. I. Basic algorithm and convergence," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 861–874, Dec. 1999.

[17] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

[18] B. Johansson, P. Soldati, and M. Johansson, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Aeras Commun.*, vol. 24, no. 8, pp. 1535–1547, Aug. 2006.

[19] B. Johansson and M. Johansson, "Distributed non-smooth resource allocation over a network," in *Proc. 48th IEEE Conf. Decis. Control*, Shanghai, China, 2009, pp. 1678–1683.

[20] H. Lakshmanan and D. P. de Farias, "Decentralized resource allocation in dynamic networks of agents," *SIAM J. Optim.*, vol. 19, no. 2, pp. 911–940, 2008.

[21] A. Cherukuri, S. Martinez, and J. Cortes, "Distributed, anytime optimization in power-generator networks for economic dispatch," in *Proc. Amer. Control Conf.*, Portland, OR, USA, 2014, pp. 172–177.

[22] T.-H. Chang, A. Nedic, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1524–1538, Jun. 2014.

[23] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 482–497, Jan. 2015.

[24] T.-H. Chang, "A proximal dual consensus ADMM method for multi-agent constrained optimization," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3719–3734, Jul. 2016.

[25] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multiagent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[26] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1131–1146, May. 2014.

[27] M. Zhu and S. Martinez, "On distributed convex optimization under inequality and equality constraints," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 151–164, Jan. 2012.

[28] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 592–606, Mar. 2012.

[29] A. Nedic and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 601–615, Mar. 2015.

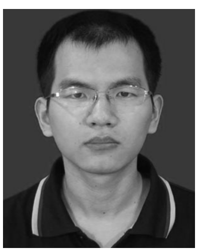
[30] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Push-sum distributed dual averaging for convex optimization," in *Proc. IEEE Conf. Decis. Control*, 2012, pp. 5453–5458.

[31] K. Cai and H. Ishii, "Average consensus on arbitrary strongly connected digraphs with time-varying topologies," *IEEE Trans. Autom. Control*, vol. 59, no. 4, pp. 1066–1071, Apr. 2014.

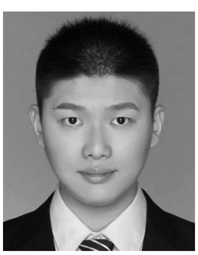
[32] D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Belmont, MA, USA: Athena Scientific, 1996.

[33] K. Cai and H. Ishii, "Quantized consensus and averaging on gossip digraphs," *IEEE Trans. Autom. Control*, vol. 56, no. 9, pp. 2087–2100, Sep. 2011.

[34] P. Samadi, H. Mohsenian-Rad, R. Schober, and V. W. S. Wong, "Advanced demand side management for the future smart grid using mechanism design," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1170–1180, Sep. 2012.



Yun Xu received the Bachelor's degree from the College of Biomedical Engineering and Instrument Science, Zhejiang University, Hangzhou, China, in 2010 and the Master's degree from the College of Automation Engineering, University of Electronic Science and Technology of China, Chengdu, China. He is currently working toward the Ph.D. degree in control theory and control engineering in the College of Electrical Engineering, Zhejiang University. His research interests include consensus and distributed optimization.

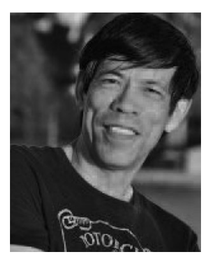


Tingrui Han received the Bachelor's degree in automation from Zhejiang University, Hangzhou, China, in 2012. He is currently working toward the Ph.D. degree in control theory and control engineering in the College of Electrical Engineering, Zhejiang University. His research interests include multiagent systems, networked control, and distributed algorithms.

Kai Cai received the B.Eng. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2006, the M.A.Sc. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2008, and the Ph.D. degree in systems science from Tokyo Institute of Technology, Tokyo, Japan, in 2011. He is currently an Associate Professor in Osaka City University, Osaka, Japan; preceding this position, he was an Assistant Professor in the University of Tokyo, in 2013 and 2014, and a Postdoctoral Fellow in the University of Toronto, from 2011 to 2013. His research interests include distributed control of multiagent systems, distributed control of discrete-event systems, and control architecture of complex networked systems.

Zhiyun Lin (SM'10) received the Bachelor's degree in electrical engineering from Yanshan University, Qinhuangdao, China, in 1998, the Master's degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2001, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, 2005. He is currently a Professor in the School of Automation, Hangzhou Dianzi University, Hangzhou, China. Preceding to this position, he held a Professor position at Zhejiang University from 2007 to 2016, and a Postdoctoral Fellow position at the University of Toronto, from 2005 to 2007. His research interests include distributed control, estimation and optimization, cooperative control of multiagent systems, hybrid control system theory, and robotics.

Gangfeng Yan received the Bachelor and Master degrees in control theory and control engineering from Zhejiang University, Hangzhou,China, in 1981 and 1984, respectively. He is currently a Professor in the College of Electrical Engineering, Zhejiang University. His research interests include hybrid systems, neural networks, and cooperative control.



Minyue Fu (F'03) received the Bachelor's degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 1982, and the M.S. and Ph.D. degrees in electrical engineering from the University of Wisconsin-Madison, Madison, WI, USA, in 1983 and 1987, respectively. From 1987 to 1989, he served as an Assistant Professor in the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI, USA. He joined the Department of Electrical and Computer Engineering, the University of Newcastle, Australia, in 1989. He is currently a Chair Professor in electrical engineering. He was a Visiting Associate Professor at the University of Iowa during 1995–1996, a Senior Fellow/Visiting Professor at Nanyang Technological University, Singapore, 2002, and a visiting Professor at Tokyo University in 2003. He has held a ChangJiang Visiting Professorship at Shandong University, a visiting Professorship at South China University of Technology, and a Qian-ren Professorship at Zhejiang University in China. His main research interests include control systems, signal processing and communications. His current research projects include networked control systems, smart electricity networks, and super-precision positioning control systems. He has been an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, *Automatica*, IEEE TRANSACTIONS ON SIGNAL PROCESSING, and *Journal of Optimization and Engineering*.