IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 13, 2019, accepted August 26, 2019, date of publication September 2, 2019, date of current version September 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2939018

Distributed Constrained Consensus of Utilities via a Self Evaluation Approach

XIAOCHU WANG[1], CHANGHAO SUN[1], AND TING SUN[2]

[1]Qian Xuesen Laboratory of Space Technology, China Academy of Space Technology, Beijing 100094, China
[2]School of Photoelectronic Information and Communication Engineering, Beijing Information Science and Technology University, Beijing 100192, China

Corresponding author: Changhao Sun (sunchanghao@qxslab.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61701498, Grant 61703403, and Grant 61605099, and in part by the Young Elite Scientists Sponsorship Program by the CAST under Grant 2016QNRC001.

ABSTRACT This paper investigates the problem of distributed constrained consensus of utilities for multiple networked nodes, which is essentially challenging in view of the fact that local constraints may hinder the global optimization across the network. Focusing on this problem, a generalized framework is developed via a self evaluation approach, in which each node uses utility evaluations to update individual states by interacting with neighbors only. Following the framework, the distributed constrained consensus algorithms is derived, with theoretical analyses presented to show the convergence and stability. In addition, an approximation method is proposed as well for ease of engineering implementation. Finally, illustrative examples are provided and analyzed to substantiate the efficacy of the proposed algorithm.

INDEX TERMS Constrained consensus, distributed optimization, heterogeneous constraints, self-evaluation, utility consensus.

I. INTRODUCTION

Networked distributed systems have been attracting great interests due to the advantage that a group of simple networked nodes working cooperatively can perform the same applications as a complex mono system. For such systems, performance maximization (or cost minimization) arises in varieties of applications, such as resource allocation, task scheduling, burden distribution, distributed estimation and motion planning [1]–[11]. In most instances, the problem of performance maximization can be equivalent to the problem of constrained consensus. As defined by Problem Eq. (1), the objective is to coordinate a series of local utility functions known to individual nodes towards consensus under constraints. Recently, increasing attention has been focused on this problem, among which distributed algorithms are considered to be more feasible and flexible than centralized ones as network scales grow.

Earlier research mainly concentrates on unconstrained consensus problems, where states of nodes are allowed to be processed arbitrarily without any constraints, to study fundamental rules for guaranteeing information among nodes to achieve a consensus in a distributed manner [1], [12]–[14]. Convergence characteristics are also analyzed under various conditions, such as higher order dynamics, time-varying topologies, information delays, communication qualities, and so on [15]–[18]. Research reveals that neighboring interaction and local agreement can be utilized as important principles for global consensus.

In more realistic applications restricted by actual conditions, states of nodes cannot get infinitely large or infinitely small, and they are subject to constraints in normal. The presence of local constraints may significantly hinder the interaction of nodes in traditional distributed algorithms and make optimization fall into local minima. To this end, the problem of constrained consensus arises and becomes a research challenge. In [2], a projected consensus algorithm is presented for the convex constrained optimization where the state of each node is restricted to lie in a distinct convex set. For achieving a constrained consensus, each node is designed to optimize its state by combining the states received from its neighbors, by taking a subgradient step to minimize its objective function, and by projecting on its individual constraint set. It is shown that, with a properly-designed stepsize that converges to zero fast enough, the states can be refined towards an optimal solution. Based on the subgradient method, some excellent distributed optimization algorithms are developed from varieties of perspectives, working on improving

The associate editor coordinating the review of this article and approving it for publication was Sotirios Goudos.

convergence accuracy, relaxing stepsize requirement, accelerating convergence rate and so on [19]–[23]. In [24], [25], a stochastic theory based projection method with Gaussian assumptions is also used to evaluate ambiguous local utilities instead of exact gradients. These works extend the algorithm conditions to milder ones. However, the subgradient based algorithms can hardly guarantee a conservative summation of the states, because the summation of state derivations is difficult to keep unbiased. Besides, the stepsizes require an elaborately design for both the convergence and the equilibrium, which in turn destroys the algorithm ability to respond to dynamical problem inputs. The primal-dual idea, which reformulates the primal constrained consensus problem properly to a dual unconstrained one, can be borrowed to develop distributed algorithms. In [6], [26], [27], distributed subgradient algorithms for reformulated dual problems are designed. In [9], a distributed algorithm is proposed based on a gradient push-sum method to solve an equivalently dual problem for constrained optimization. Although excellent work has been presented to solve constrained consensus problems, these algorithms cannot be used to solve Problem Eq. (1) because their asymmetric formed equations will cause the nonconservation of the state summation. Meanwhile, they often require an elaboratedly-designed sequence of stepsizes to guarantee convergence. This, however, increases the difficulty in designing algorithms, imposes a constraint on the algorithm universality, and loses the rapid responsibility to the problems with varying parameters. The event-triggered strategy is used in some research to solve a network optimization problem, with requiring that all states be strictly equal to each other [28], [29]. Each node in this strategy refines itself according to the sample-data that is periodically triggered to update. This strategy works well and is inspiring, but it has not considered the individual constraints or the summation constraint. As an important branch, game theory based algorithms are often used for distributed optimization by means of neighboring negotiation. However, global convergence is not guaranteed by these algorithms, and only Nash equilibria can be obtained [30]. To the best knowledge of the authors, the most effective algorithms for solving our problem are the barrier-based Lagrange methods [8], [31]. In [8], a barrier function is employed to develop a constrained consensus algorithm with constraints of lower boundaries. In [31], likewise, a so-called $\theta-$ logarithmic barrier is introduced to reformulate the problem and then a corresponding algorithm based on the distributed interior point is proposed for the dual problem by using the Lagrange function (see i.e., Eq. (18)). It is shown that, with a properly increasing parameter θ, consensus can be almost achieved with the states strongly restricted within constraints. However, this logarithmic barrier based algorithm shows a deep sensitivity to stepsizes and boundary detections. Meanwhile, the reformulated problem imposes an additional constraint on the variable positivity of logarithmic functions. However, once the states are overshot and become infeasible (i.e., running out of the constraints) in a single iteration, the algorithm will run out immediately and all the rest iterations will go wrong. Besides, even given perfect parameters and conditions, these algorithms can never achieve the exact global optimum, because the exact global optimum is the algorithm singular point that is not allowed by mathematics.

We should point out that, for the sake of brevity, our survey of previous work here is short of complete and only contains the previous work related to our work. For a more complete survey we refer to [22], [26], [31] and the references therein.

In this paper, we consider a novel, concise, and nature-inspired framework along with a derived algorithm for solving a class of general state-constrained consensus problems in a fully distributed way. As a further research of the previous work [32] that only targets the simplified linear homogeneous problem of state consensus under upper bounded constraints, this paper solves the more general problem where nonlinear heterogeneous utilities are allowed and lower bounded constraints exist as well. This makes sense because most similar problems in actual life are not ideally linear or homogeneous. Although this extension may seem trivial, the nontrivial novelty lies exactly in the improved uniform framework that guarantees global convergence, and in the ingenious way by which we package the nonlinear heterogeneous utilities into the uniform framework. In particular, the significant contributions are threefold.

1) A novel principle following ‘‘the law of connected vessels’’ in Physics is developed to help design distributed algorithms, where no Lagrange functions or dual methods are needed to reformulate the primal problem, and where a self evaluation approach is proposed to guide nodes to interact. This principle has great potential to inspire and simplify the algorithm designs for similar problems. Furthermore, the proposed principle is extended to a mild condition so that no time-varying global information is required and all parameters are invariant, making the principle fully distributed and be able to respond to the problems with varying inputs.
2) Existing related algorithms have kinds of flaws that hinder their applications. They may suffer from stepsizes (see e.g., [2], [19]–[25], [31]), infeasible solutions and algorithm errors (see e.g., [6], [9], [26], [27], [31]), and local optima (see e.g., [8], [31]). The algorithm proposed in this paper can overcome all these drawbacks. Firstly, our continuous formed algorithm is free of stepsizes. But even if our algorithm is transformed to a discrete form by common discrete methods, no sequence of variable stepsizes will be required to guarantee convergence, and a proper constant stepsize will work. Therefore, the effectiveness of our algorithm is time-invariant. Secondly, infeasible overshot solutions would never make our algorithm run out, because our algorithm has no singular point. Thirdly, our algorithm can achieve an exact global optimum. By using our algorithm, any local optimum will be unstable, and the solution will converge to the global optimum.

3) To handle the noncontinuous switching problem at boundaries of constraints, which is a common issue that embarrasses numerical calculations and engineering applications severely (see e.g., [8], [31]), an approximation method is proposed to help nodes evaluate local utilities in a smooth way, weakening the implementation requirement of the algorithm greatly.

The remainder of this paper is organized as follows. In Section II, notations, constraints definitions and the problem formulation are introduced. In Section III, the novel framework of distributed constrained consensus is proposed, which is inspired by "the law of connected vessels". In Section IV, Algorithm 1 is proposed for utility consensus with state constraints, and an approximation method is provided as well. In Section V, illustrative examples and the comparison with the benchmark algorithm are given. Finally, Section VI concludes this paper.

II. PRELIMINARIES

It is common to model the information interaction among networked nodes by a graph $\mathcal{G}$, which can be defined by a 2-tuple pair $(\mathcal{N}, \mathcal{E})$. $\mathcal{N} = \{1, \dots, n\}$ is a finite nonempty node set and $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ is an edge set of ordered pairs of nodes. An edge (i, j) in a directed graph means that node j can receive information from node i, but not necessarily vice versa. The adjacency matrix $\mathcal{A} = [a_{ij}]_{n\times n}$ of a weighted directed graph is defined as $a_{ii} = 0$ and $a_{ij} > 0$ if $(j, i) \in \mathcal{E}$, where $i \neq j$. Specially, an undirected graph is said to be balanced when $a_{ij} = a_{ji}$ holds for any $i \neq j$. Node j is said to be one neighbor of node i if $a_{ij} > 0$. Moreover, a neighbor set $\mathcal{N}_i$ is defined to contain all the neighbors of node i, such that $a_{ij} > 0$ holds for any $j \in \mathcal{N}_i$, $\forall i \in \mathcal{N}$.

A directed graph is said to contain a spanning tree if there is a path consisting of a set of sequential edges that start from a certain node in the graph and connect all the other nodes. A connected graph is an undirected graph containing a spanning tree so that any two nodes are connected to each other by paths. A component of an undirected graph $\mathcal{G}$ is a subgraph in which any two nodes are connected to each other nodes, and which is connected to no additional nodes in $\mathcal{G}$. For a connected undirected graph, cut-vertices refer to a set of nodes whose removal increases the number of components.

Denote by $x_i = x_i(t)$ the state of node i, and by $f_i = f_i(x_i)$ the corresponding utility. The assignment of states of all the nodes is denoted by $\mathcal{X} = \mathcal{X}(t) = \{x_1(t), \cdots, x_n(t)\}$. For practical problems, three constraints are assumed as follows in this paper, and the assignment feasibility is defined.

Constraint 1 (Lower boundedness): States are assumed to be constrained by a series of constant lower bounds $\{x_{i,MIN}\}$. *i.e., for all* $t \geq 0$, $x_i \geq x_{i,MIN}$ *holds for any* $i \in \mathcal{N}$. *Specially, we only focus on the case where* $x_{i,MIN} \geq 0$, *because any case with* $x_{i,MIN} < 0$ *can always be transformed to it after being translated/shifted by proper offsets.*

Constraint 2 (Upper boundedness): Capacities of nodes are assumed to be finite and constrained by a series of constant upper bounds $\{x_{i,MAX}\}$, *i.e., for all* $t \geq 0$, $x_i \leq x_{i,MAX}$ *holds for any* $i \in \mathcal{N}$. *Note that* $x_{i,MAX} \geq x_{i,MIN} \geq 0$.

Constraint 3 (Summation immutability): The summation of states across the entire network is required to be conservative and has immutability, i.e., if $\sum_{i\in\mathcal{N}} x_i(0) = x_D$, *then for* $\forall t > 0$, $\sum_{i\in\mathcal{N}} x_i(t) \equiv x_D$.

Definition 1 (Assignment feasibility): An assignment $\mathcal{X}$ *is said to be feasible, only if the states of nodes are lower-bounded, upper-bounded and summation immutable simultaneously. Define* $\mathbb{X}_F$ *as the set that contains all the feasible assignments (for a certain problem).*

Therefore, the constrained consensus problem can be concluded as **finding/optimizing a feasible assignment targeting the consensus of utilities under constraints**, i.e.,

$$\textbf{Optimize } \mathcal{X}^* = \arg\min_{\mathcal{X}\in\mathbb{X}_F} \sum_{i,j\in\mathcal{N}} |f_i - f_j| \tag{1a}$$

$$\textbf{s.t.} \sum_{i\in\mathcal{N}} x_i = x_D; \quad x_{i,MIN} \leq x_i \leq x_{i,MAX}, \ \forall i \in \mathcal{N}. \tag{1b}$$

In this paper, for each node i, the node utility f_i is assumed to be a strictly monotonic differentiable function of x_i in the definition domain, with $\frac{\partial f_i}{\partial x_i} > 0$.

III. FRAMEWORK OF DISTRIBUTED OPTIMAL CONSTRAINED CONSENSUS

A. INSIGHT AND INSPIRATION

Ordinary unconstrained consensus would fail in optimizing allocation problems with constraints. To explain the reason behind this, Fig. 1 is taken as an example to highlight the effects of the presence of local constraints, where $x_\mathrm{A} = x_\mathrm{B} = \cdots = x_\mathrm{E} = 2 \neq x_\mathrm{G} = x_\mathrm{H} = \cdots = x_\mathrm{L} = 100$, $x_\mathrm{F} = x_{\mathrm{F},MAX} = 1$. The mechanism in ordinary unconstrained consensus algorithms is to drive the state of each node towards the mean value of its neighbors. However, since x_F has reached its boundary $x_{\mathrm{F},MAX}$, x_F cannot increase, although $x_\mathrm{F} < x_\mathrm{E}$ and $x_\mathrm{F} < x_\mathrm{I}$. As a result, x_I cannot decrease, either (otherwise, the conservative summation will be destroyed). Likewise, x_E cannot increase. It is then clear

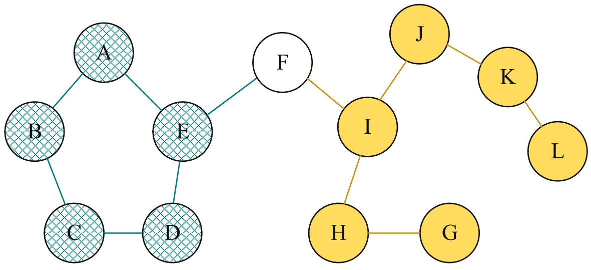

FIGURE 1. Example of a constrained resource allocation. Suppose all nodes have unconstrained boundaries except for node F, whose upper boundary is constrained by $x_{\mathrm{F},MAX} = 1$. Initial states are given as $x_\mathrm{A} = x_\mathrm{B} = \cdots = x_\mathrm{E} = 2$, $x_\mathrm{F} = 1$, and $x_\mathrm{G} = x_\mathrm{H} = \cdots = x_\mathrm{L} = 100$. It can be seen that node F acts as a cut-vertex that cuts the graph into two independent components, and significantly blocks the interaction across it under ordinary unconstrained consensus algorithms. As a result, the left and right components will achieve consensus on two local agreements separately, but cannot achieve a global optimization.

that the consensus is trapped in a local minimum even if the two components on both sides of node F have not achieved a global agreement. In general, since the flow across the network may be blocked by the saturated nodes that have reached their boundaries and whose states can not increase or decrease anymore, ordinary algorithms would only lead the optimization to local optima.

Towards global optimization, a novel principle of interaction is required. As a natural law, "the law of connected vessels" in Physics is enlightening in developing the principle. As shown in Fig. 2, the connected-vessel system consists of three vessels, among which only B is covered by a lid while neither A nor C is. Suppose that the vessel B is full of water initially. Although water tends to flow from high level to low level in normal, the water in Fig. 2 will actually flow from B towards C, regardless of that the water level in B is lower than that in C. As a matter of fact, the water pressure in B has no relation with its own water level, but is equivalent to that in A, which is the maximal water level of the system. Therefore, the example system would get a constrained consensus finally. Inspired by this natural law, one can design a novel framework.

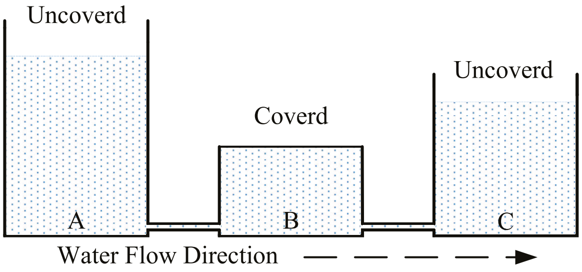

FIGURE 2. A three-connected-vessel system with the vessel B being covered by a lid. Given such an initial state, the water will flow across B, along the direction from A to C.

B. FUNDAMENTAL PRINCIPLE

The nature-inspired principle for solving constrained consensus can be concluded as follows

1) Each node should update its state not by comparing its utility with those of neighbors, but by comparing the "utility evaluations".
2) For every individual node, its utility evaluation should be such a piecewise function that the evaluation equals to the its utility only when the state of node doesn't touch the boundaries of constraints, and equals to some properly-designed values otherwise.

C. NOVEL UNIFORM FRAMEWORK

The above principle could yield various frameworks for solving constrained consensus problems. In this part, we propose one such framework as the basis of the formal algorithm.

First of all, we simplify Problem Eq. (1) by assuming $f_i = x_i$, and then the full problem will be considered in Section IV.

Before going on, some denotations are made for concision. With respect to the relationship between the state and the boundaries, nodes are clustered into three dynamic status sets, termed as $\mathcal{N}_\odot$, $\mathcal{N}_\ominus$, and $\mathcal{N}_\oplus$, respectively:

- $i \in \mathcal{N}_\odot$ if $x_{i,MIN} < x_i < x_{i,MAX}$,
- $i \in \mathcal{N}_\ominus$ if $x_i = x_{i,MIN}$, and
- $i \in \mathcal{N}_\oplus$ if $x_i = x_{i,MAX}$.

Denote the evaluation of x_i by $\hat{x}_i$ for each node. Refer to the (time-varying) maximal state among nodes as $\bar{x}$, and to the minimal state among nodes as $\underline{x}$, respectively, i.e., $\bar{x} = \max\{x_1, x_2, \ldots, x_n\}$, and $\underline{x} = \min\{x_1, x_2, \ldots, x_n\}$. Temporarily, $\bar{x}$ and $\underline{x}$ are assumed to be available to every node. With this assumption, each node obtains its evaluation $\hat{x}_i$ locally following

$$\hat{x}_i = \begin{cases} \underline{x} & \text{if } i \in \mathcal{N}_\ominus \\ x_i & \text{if } i \in \mathcal{N}_\odot \\ \bar{x} & \text{otherwise } (i \in \mathcal{N}_\oplus) \end{cases} \tag{2}$$

The framework is then proposed as

$$\dot{x}_i = -\gamma \sum_{j \in \mathcal{N}_i} a_{ij}(\hat{x}_i - \hat{x}_j) \tag{3}$$

where γ is a positive scaler.

To see the convergence of the framework, some lemmas are needed.

Lemma 1: Framework Eq. (3) keeps lower boundedness, i.e., if $x_i(0) \geq x_{i,MIN}$ for any $i \in \mathcal{N}$, then for any $t > 0$, $x_i(t) \geq x_{i,MIN}$.

Proof: Suppose that, without loss of generality, the node set $\mathcal{N}_\ominus$ is empty until a certain moment $t = T_k$. Therefore, for any time $t < T_k$, all nodes $i \in \mathcal{N}$ keep lower boundedness.

At $t = T_k$, it follows from Eq. (2) that $\hat{x}_i = \underline{x}$ holds for every $i \in \mathcal{N}_\ominus$ since $x_i = x_{i,MIN}$. Meanwhile, for any $i \notin \mathcal{N}_\ominus$, $\hat{x}_i \geq \underline{x}$. Hence, for any $i \in \mathcal{N}_\ominus$, one can yield the derivative of x_i along Eq. (3) and obtain $\dot{x}_i = -\gamma \sum_{j \in \mathcal{N}_i} a_{ij} \left(\underline{x} - \hat{x}_j\right) \geq 0$. It means that for any node with $x_i = x_{i,MIN}$, its state x_i will stop decreasing immediately.

Considering the arbitrariness of T_k, one can conclude that for any $t > 0$, $x_i(t) \geq x_{i,MIN}$ holds if $x_i(0) \geq x_{i,MIN}$. This completes the proof. □

Lemma 2: Framework Eq. (3) keeps upper boundedness, i.e., if $x_i(0) \leq x_{i,MAX}$ for any $i \in \mathcal{N}$, then for any $t > 0$, $x_i(t) \leq x_{i,MAX}$.

Proof: This proof is similar to that of Lemma 1. Without loss of generality, suppose that the node set $\mathcal{N}_\oplus$ is empty until a certain moment $t = T_k$. Therefore, for any time $t < T_k$, all nodes $i \in \mathcal{N}$ keep upper-boundedness.

At $t = T_k$, it follows from Eq. (2) that $\hat{x}_i = \bar{x}$ holds for any $i \in \mathcal{N}_\oplus$. Meanwhile, for any $i \notin \mathcal{N}_\oplus$, $\hat{x}_i \leq \bar{x}$. Hence, for any $i \in \mathcal{N}_\oplus$, the derivative of x_i along Eq. (3) yields $\dot{x}_i \leq 0$. It means that for any node with $x_i = x_{i,MAX}$, its state x_i will stop increasing immediately.

Considering the arbitrariness of T_k, one can conclude that for any $t > 0$, $x_i(t) \leq x_{i,MAX}$ holds if $x_i(0) \leq x_{i,MAX}$. This completes the proof. □

Lemma 3: Framework Eq. (3) is conservative and keeps summation immutability for a balanced $\mathcal{G}$, i.e., if $\sum_{i\in\mathcal{N}} x_i(0) = x_D$, then for any $t > 0$, $\sum_{i\in\mathcal{N}} x_i(t) = x_D$.

Proof: For any time $t \geq 0$, we consider the growth rate $\dot{x}_i$ of all nodes along the trajectory of our algorithm, which yields

$$\begin{aligned}
&\sum_{i\in\mathcal{N}} \dot{x}_i \\
&\quad = \sum_{i\in\mathcal{N}} \Big[-\gamma \sum_{j\in\mathcal{N}_i} a_{ij}(\hat{x}_i - \hat{x}_j)\Big] \\
&\quad = -\gamma \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} a_{ij}(\hat{x}_i - \hat{x}_j) \\
&\quad = -\frac{1}{2}\gamma \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} a_{ij}(\hat{x}_i - \hat{x}_j) - \frac{1}{2}\gamma \sum_{j\in\mathcal{N}} \sum_{i\in\mathcal{N}} a_{ji}(\hat{x}_j - \hat{x}_i) \\
&\quad = -\frac{1}{2}\gamma \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} a_{ij}(\hat{x}_i - \hat{x}_j) + \frac{1}{2}\gamma \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} a_{ij}(\hat{x}_i - \hat{x}_j) \\
&\quad = 0. \qquad\qquad (4)
\end{aligned}$$

Here we have used Eq. (3) to obtain the first equality, have used the fact $\sum_{j\in\mathcal{N}_i} a_{ij}(\cdot) \equiv \sum_{j\in\mathcal{N}} a_{ij}(\cdot)$ to obtain the second equality, have switched the symbols of i and j to obtain the third equality, and have used the fact that $a_{ij} = a_{ji}$ and have switched the order of the summation signs to obtain the fourth equality. Given the initial condition $\sum_{i\in\mathcal{N}} x_i(0) = x_D$, we complete the proof. □

Lemma 4 ([12]): Suppose that a networked system consisting of node set $\mathcal{N}$ is described by a graph $\mathcal{G}$. If $\mathcal{G}$ is undirected connected, $\gamma > 0$, and all x_i have the same variable domain, then with the protocol $\dot{x}_i = -\gamma \sum_{j\in\mathcal{N}} a_{ij}\left(x_i - x_j\right)$, $x_i \to x_j$ holds for any i and j as $t \to \infty$.

Theorem 1: With Eq. (2), Framework Eq. (3) drives all x_i in a connected $\mathcal{G}$ to a constrained consensus with x_i both lower and upper bounded. In addition, if $\mathcal{G}$ is balanced, all x_i keep summation immutability.

Proof: At any arbitrary time $t > 0$, the situation of states x_i must belong to one of the following two cases: 1) $\mathcal{N}_\odot = \mathcal{N}$, and 2) $\mathcal{N}_\odot \neq \mathcal{N}$. At first, we look into these cases respectively. Note that for any i, $\hat{x}_i$ may not be smooth but would be differentiable almost everywhere. By using nonsmooth analysis [13], [33] and terming differential inclusions [34] by the symbol "$\overset{a.e.}{=}$" with a.e. standing for "almost everywhere", the convergence of the state evaluations $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n$ is studied as follows.

Case 1 ($\mathcal{N}_\odot = \mathcal{N}$). Since $\mathcal{N}_\odot = \mathcal{N}$, it is clear that for all $i \in \mathcal{N}$, $\hat{x}_i = x_i$ holds according to Eq. (2). Therefore, Framework Eq. (3) can be rewritten as $\dot{\hat{x}}_i = -\gamma \sum_{j\in\mathcal{N}_i} a_{ij}(\hat{x}_i - \hat{x}_j)$. It then follows from Lemma 1 that all $\hat{x}_i$ will be driven asymptotically to achieve a consensus, i.e., $\hat{x}_i \to \hat{x}_j$, as $t \to \infty$.

Case 2 ($\mathcal{N}_\odot \neq \mathcal{N}$). Since $\mathcal{N}_\odot \neq \mathcal{N}$, it implies that $\mathcal{N}_\ominus \cup \mathcal{N}_\oplus \neq \emptyset$. Without loss of generality, assume that $\mathcal{N}_\ominus \neq \emptyset$ and $\mathcal{N}_\oplus \neq \emptyset$. Besides, the existence of a common overlap among the state domains of all the nodes is assumed for concise expression (see Remark 1). Refer to the very node whose state is the maximum as node k_1, and refer to the very node whose state is the minimum as node k_2, i.e., $\bar{x} = x_{k_1}$, $\underline{x} = x_{k_2}$ (Actually, there can be more than one node that has the same maximal state. In this case, k_1 can be designated as a combined node that consists of the set of all the nodes with the maximal state. Likewise, k_2 can be designated as a combined node that consists of the set of all the nodes with the minimal state. By making such a designation, one can always convert the analysis to that with single node). According to Eq. (2), one has that $\hat{x}_i = x_{k_1} = \hat{x}_{k_1}$ for $i \in \mathcal{N}_\oplus$, $\hat{x}_i = x_{k_2} = \hat{x}_2$ for $i \in \mathcal{N}_\ominus$, and $\hat{x}_i = x_i$ for $i \in \mathcal{N}_\odot$. It is then clear that, by denoting $\Omega = \mathcal{N}_\ominus \cup \{k_1\} \cup \{k_2\}$, Framework Eq. (3) yields the updating protocol for $\hat{x}_i$, following

$$\dot{\hat{x}}_i \overset{a.e.}{=} -\gamma \sum_{j\in\mathcal{N}_i} a_{ij}(\hat{x}_i - \hat{x}_j), \quad i \in \Omega \qquad (5a)$$

$$\hat{x}_i = \hat{x}_{k_1}, \quad i \in \mathcal{N}_\oplus \qquad (5b)$$

$$\hat{x}_i = \hat{x}_{k_2}, \quad i \in \mathcal{N}_\ominus. \qquad (5c)$$

With the equations of Eqs. (5b) and (5c), Eq. (5a) can be rewritten as

$$\begin{aligned}
\dot{\hat{x}}_i &\overset{a.e.}{=} -\gamma \sum_{j\in\mathcal{N}_i} a_{ij}(\hat{x}_i - \hat{x}_j) \\
&= -\gamma \sum_{j\in\mathcal{N}} a_{ij}(\hat{x}_i - \hat{x}_j) \\
&= -\gamma \sum_{j\in\mathcal{N}_\odot} a_{ij}(\hat{x}_i - \hat{x}_j) \\
&\quad -\gamma \sum_{j\in\mathcal{N}_\oplus} a_{ij}(\hat{x}_i - \hat{x}_{k_1}) - \gamma \sum_{j\in\mathcal{N}_\ominus} a_{ij}(\hat{x}_i - \hat{x}_{k_2}) \\
&= -\gamma \sum_{j\in\mathcal{N}_\odot} a_{ij}(\hat{x}_i - \hat{x}_j) \\
&\quad -\gamma\Big(\sum_{j\in\mathcal{N}_\oplus} a_{ij}\Big)(\hat{x}_i - \hat{x}_{k_1}) \\
&\quad -\gamma\Big(\sum_{j\in\mathcal{N}_\ominus} a_{ij}\Big)(\hat{x}_i - \hat{x}_{k_2}) \\
&= -\gamma \sum_{j\in\mathcal{N}_\odot} a_{ij}(\hat{x}_i - \hat{x}_j) \\
&\quad -\gamma\Big(\sum_{k\in\mathcal{N}_\oplus} a_{ik}\Big)(\hat{x}_i - \hat{x}_{k_1}) \\
&\quad -\gamma\Big(\sum_{k\in\mathcal{N}_\ominus} a_{ik}\Big)(\hat{x}_i - \hat{x}_{k_2}) \\
&= -\gamma \sum_{j\in\Omega} \tilde{a}_{ij}(\hat{x}_i - \hat{x}_j). \qquad (6)
\end{aligned}$$

Here

i) $\tilde{a}_{ij} = a_{ij}$ for any $i, j \in \mathcal{N}_\odot$;

ii) $\tilde{a}_{ik_1} = a_{ik_1} + \sum_{k\in\mathcal{N}_\oplus} a_{ik} \geq a_{ik_1}$ for any $j \in \mathcal{N}_\oplus$, $\tilde{a}_{k_1 i} = a_{k_1 i}$ for any $i \in \Omega$;

iii) $\tilde{a}_{ik_2} = a_{ik_2} + \sum_{k \in \mathcal{N}_\ominus} a_{ik} \geq a_{ik_2}$ for any $j \in \mathcal{N}_\ominus$, $\tilde{a}_{k_2 i} = a_{k_2 i}$ for any $i \in \Omega$.

Therefore, if $\mathcal{G}(\mathcal{A})$ is undirected connected, $\tilde{\mathcal{G}}(\tilde{\mathcal{A}})$ will be also undirected connected. With the notice that the domains of $\hat{x}_i$ are the same for all i, it then follows from Lemma 4 that $\hat{x}_i \to \hat{x}_j$ asymptotically as $t \to \infty$ for any $i, j \in \Omega$. Once again, since $\hat{x}_i = \hat{x}_{k_1}$ for $i \in \mathcal{N}_\oplus$ and $\hat{x}_i = \hat{x}_{k_2}$ for $i \in \mathcal{N}_\ominus$, one finally obtains that, for all nodes $i, j \in \mathcal{N}$, $\hat{x}_i \to \hat{x}_j$ asymptotically as $t \to \infty$.

One can conclude that the convergence of each case is consistent such that $\hat{x}_i \to \hat{x}_j$ asymptotically as $t \to \infty$ almost all the time, for $\forall i, j \in \mathcal{N}$. This implies in turn that $\bar{x} \to \underline{x}$ as $t \to \infty$ under constraints. The converged equilibrium will be $\hat{x}_i = \hat{x}_j$ almost all the time. Observing Eq. (3) with this knowledge, one would obtain $\dot{x}_i = 0$ for all i at the equilibrium, implying that the equilibrium is stable. Furthermore, with respect to the relationship between $\hat{x}_i$ and x_i, one can also obtain the equivalent converged equilibrium that, for all nodes, $x_i = x_{\text{mean}} \triangleq \text{mean}\{x_j | j \in \mathcal{N}_\odot\}$ for any $i \in \mathcal{N}_\odot$, $x_i = x_{i,MIN} \geq x_{mean}$ for any $i \in \mathcal{N}_\ominus$, and $x_i = x_{i,MAX} \leq x_{\text{mean}}$ for any $i \in \mathcal{N}_\oplus$. It is obvious that, by making an observation to $V = \sum_{i,j \in \mathcal{N}} |x_i - x_j|$, the equilibrium strictly corresponds to the goal of constrained consensus because V reaches the minimum at this equilibrium, upon which any perturbation to the states would cause an increase of V. Therefore, the proof is completed. □

Remark 1: It is worthy of noting that the proof is processed in the sense of "almost everywhere" and under the assumption of the existence of a common overlap among the domains of all the nodes. Actually, exceptions may occur in the case where no common overlap exists. In this situation, the statement $\hat{x}_i \to \hat{x}_j$ would not be always true, because $\lim_{t \to \infty} (\hat{x}_i - \hat{x}_j)$ does not always equal to 0 but may equal to a constant that is determined by the constraints, and in this case, the symbol "$\to$" stands for "is driven towards" instead. However, the global convergence of the proposed framework would not suffer from the possible exceptions, because $\hat{x}_i$ is always driven towards $\hat{x}_j$ almost everywhere, except for some finite time instants that occur momentarily along with the continuous time progress (that consists of infinite time instants). For example, consider a special two-node system consisting of A *and* B *with constraints $0 \leq x_{\text{A}} \leq 2$ and $3 \leq x_{\text{B}} \leq 5$, and given the initial values $x_{\text{A}}(0) = 1$ and $x_{\text{B}}(0) = 4$. Although the constrained consensus made by the proposed framework is $\lim_{t \to \infty} (\hat{x}_{\text{B}} - \hat{x}_{\text{A}}) = 1 \neq 0$, the global convergence remains such that x_{A} and x_{B} are driven towards each other almost all the time under constraints. Since the exceptions do not matter in analyzing the global convergence, the proof is performed under general situations for making the expression concise.*

The mechanism of the proposed framework can be illustrated by taking Fig. 1 as an example once again. In the beginning, since node F reaches its upper boundary as $x_{\text{F}} = 1$, one obtains $\hat{x}_{\text{F}} = \bar{x} = 100$ following Eq. (2). As a result, with $\hat{x}_{\text{E}} = 2$ and $\hat{x}_{\text{I}} = 100$, x_{F} decreases and x_{E} increases following Eq. (3). After that, as soon as x_{F} decreases, it becomes that $\hat{x}_{\text{F}} = x_{\text{F}} < 1 < \hat{x}_{\text{I}}$, which leads x_{I} to decrease and leads x_{F} to increase. As one can observe, the overall effect is equivalent to that the resource on node I flows to node E across the saturated node F. Since this process will continue until both sides of node F achieve a consensus, the global optimization of the constrained consensus will be made.

Note that although Framework Eq. (3) using Eq. (2) guarantees the constrained consensus, the assumption is overstrict, i.e., it's not easy for each node to track the time-varying $\bar{x}$ and $\underline{x}$ in realtime. The reason lies in that there is no model to describe the varying of $\bar{x}$ and $\underline{x}$. Therefore, tracking by using locally neighboring synchronization will be always lagging behind global signals $\bar{x}$ and $\underline{x}$. One may use an inner-loop tracking protocol to do this, but it would cost much communication and calculational bandwidths. To this end, we extend the framework to a feasible one with a mild assumption, so that each node evaluates its state following

$$\hat{x}_i = \begin{cases} \underline{x_{MIN}} & \text{if } i \in \mathcal{N}_\ominus \\ x_i & \text{if } i \in \mathcal{N}_\odot \\ \overline{x_{MAX}} & \text{otherwise } (i \in \mathcal{N}_\oplus) \end{cases} \tag{7}$$

where $\overline{x_{MAX}}$ and $\underline{x_{MIN}}$ are the maximal upper boundary and minimal lower boundary among nodes, respectively, i.e., $\overline{x_{MAX}} = \max\{x_{1,MAX}, \cdots, x_{n,MAX}\}$, $\underline{x_{MIN}} = \min\{x_{1,MIN}, \cdots, x_{n,MIN}\}$.

Theorem 2: With Eq. (7), Framework Eq. (3) drives all x_i in a connected $\mathcal{G}$ to a constrained consensus with x_i both lower and upper bounded. In addition, if $\mathcal{G}$ is balanced, all x_i keep summation immutability.

Proof: Obviously, the proof is similar with that of Theorem 1. By copying the process and replacing the symbols $\hat{x}_{k_1}$ and $\hat{x}_{k_2}$ with $\overline{x_{MAX}}$ and $\underline{x_{MIN}}$, respectively, one can obtain the proof likewise. □

Remark 2: In Theorem 2, Framework Eq. (3) using Eq. (7) relaxes the assumption of Eq. (2) to a mild requirement, where each node no longer needs to track either time-varying signal (i.e., $\bar{x}$ or $\underline{x}$). Instead, two global but static variables $\overline{x_{MAX}}$ and $\underline{x_{MIN}}$ are used for self evaluation, and each node only needs to acquire the two variables once aforehand. In particular, the acquisition can be achieved by using a traditional distributed consensus protocol, e.g., $e_i(t) = \max_{j \in \mathcal{N}_i} x_{j,MAX}$. Therefore, the framework can work in a fully distributed manner. It can be also seen that the framework has no singular point that will cause errors in mathematics.

IV. CONSTRAINED CONSENSUS ALGORITHM

Considering the problem Eq. (1), this part studies the constrained consensus algorithm that is built upon our framework.

A. ALGORITHM

The algorithm follows the basic idea of the framework such that each node uses evaluations of utilities to interact with neighbors. With $\hat{f}_i$ denoting the utility evaluation, each node i obtains $\hat{f}_i$ according to

$$\hat{f}_i = \begin{cases} \underline{f_{MIN}} & \text{if } i \in \mathcal{N}_\ominus \\ f_i & \text{if } i \in \mathcal{N}_\odot \\ \overline{f_{MAX}} & \text{otherwise } (i \in \mathcal{N}_\oplus) \end{cases} \tag{8}$$

where $\overline{f_{MAX}} = \max\{f_1(x_{1,MAX}), \cdots, f_n(x_{n,MAX})\}$, $\underline{f_{MIN}} = \min\{f_1(x_{1,MIN}), \cdots, f_n(x_{n,MIN})\}$.

With the self utility evaluation, the algorithm is proposed as

$$\dot{x}_i = -\gamma \sum_{j \in \mathcal{N}_i} a_{ij}(\hat{f}_i - \hat{f}_j) \tag{9}$$

where $\gamma > 0$.

Theorem 3: If a connected graph $\mathcal{G}$ is balanced, the algorithm Eq. (9) using Eq. (8) will optimize Problem Eq. (1) to a global optimization.

Proof: Since the connected graph $\mathcal{G}$ is balanced, one can obtain in the same way as presented in Lemmas 1-3 that by using the algorithm Eq. (9), all x_i keep lower boundedness, upper boundedness and summation immutability for any $t > 0$. This implies that the assignments are guaranteed to be optimized within $\mathbb{X}_\mathrm{F}$.

Next, one can obtain the derivative of f_i along Eq. (3) as

$$\begin{aligned} \dot{f}_i &= \frac{\partial f_i}{\partial x_i}\dot{x}_i \\ &= -\gamma \sum_{j \in \mathcal{N}_i} \frac{\partial f_i}{\partial x_i} \cdot a_{ij}(\hat{f}_i - \hat{f}_j) \\ &= -\gamma \sum_{j \in \mathcal{N}_i} \breve{a}_{ij}(\hat{f}_i - \hat{f}_j) \end{aligned} \tag{10}$$

where $\breve{a}_{ij} \triangleq \frac{\partial f_i}{\partial x_i} \cdot a_{ij}$. Since $\frac{\partial f_i}{\partial x_i} > 0$, it follows that for any $i, j \in \mathcal{N}$, $\breve{a}_{ij}$ and a_{ij} have the same sign. If $\mathcal{G}(\mathcal{A})$ is connected, $\breve{\mathcal{G}}(\breve{\mathcal{A}})$ will be connected as well.

Therefore, one can obtain from Theorem 2 and the proof of Theorem 1 that, with noting the consistency in form, Eq. (10) using Eq. (8) will drive all f_i in a connected $\breve{\mathcal{G}}$ to achieve a consensus under constraints. Since it obviously corresponds to the optimal assignment $\mathcal{X}^*$ that minimizes the maximal difference between any f_i and f_j, the proof is completed. □

Remark 3: Algorithm Eq. (9) extends Framework Eq. (3) to a more general case, where the utility is allowed to be any monotonic increasing functions of the state for each node. Likewise, this algorithm is fully distributed since the obtainment of utility evaluation works locally and relies on no realtime communication.

B. ENGINEERING APPROXIMATION

It is worthwhile to note that, due to the noncontinuous switching, using Eq. (7) and Eq. (8) to obtain evaluations still puts rigorous requirements on threshold detection and numerical accuracy for actual applications. To relax the requirements and improve the algorithm usability, an approximation for obtaining noncontinuous evaluations is presented in this part.

First, consider a two-state logical function $\sigma(u, \tau)$ and a continuous differentiable function $s(u, \tau)$, which are

$$\sigma(u) = \begin{cases} 0, & \text{if } u < \tau \\ 1, & \text{if } u \geq \tau \end{cases} \tag{11}$$

and

$$s(u, \tau) = 0.5\,[\tanh(\beta(u - \tau)) + 1] \tag{12}$$

with $\beta > 0$, respectively.

It is then clear that, as β increases, $s(u, \tau)$ will approximate to $\sigma(u, \tau)$ more closely (e.g., see Fig. 3). One may obtain this conclusion as Lemma 5.

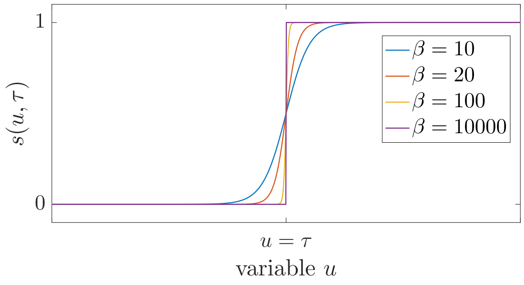

FIGURE 3. The shapes of $s(u, \tau)$ with varied β.

Lemma 5: If the parameter β is large enough, $s(u, \tau)$ can be used to approximate $\sigma(u, \tau)$ with an excellent accuracy.

Next, consider a general three-state switching function $\sigma^{3-\text{state}}(u, \tau_l, \tau_h)$ as

$$\sigma^{3-\text{state}}(u) = \begin{cases} y_1, & \text{if } u \leq \tau_l \\ y_2, & \text{if } \tau_l < u < \tau_h \\ y_3, & \text{if } u \geq \tau_h \end{cases} \tag{13}$$

with y_1, y_2, y_3 being arbitrary output quantities. Therefore, by doubly applying Lemma 5, one can obtain

$$\begin{aligned} z(u, \tau_l, \tau_h) &= (1 - \omega_1)(1 - \omega_2)y_1 + \omega_1(1 - \omega_2)y_2 + \omega_2 y_3 \\ \omega_1 &= s(u, \tau_l) \\ \omega_2 &= s(u, \tau_h) \end{aligned} \tag{14}$$

Lemma 6: If the parameter β is large enough, Eq. (14) can be used to approximate a three-state switching function $\sigma^{3-\text{state}}(u, \tau_l, \tau_h)$ with an excellent accuracy (see e.g., Fig. 4).

Finally, one can obtain the following approach to approximate $\hat{f}_i$ in a smooth continuous way.

Theorem 4: With a large enough β, z_i in Eq. (15) is an excellent approximation to $\hat{f}_i$ in Eq. (8).

$$\begin{aligned} z_i &= (1 - \omega_1)(1 - \omega_2)\underline{f_{MIN}} + \omega_1(1 - \omega_2)f_i + \omega_2\overline{f_{MAX}} \\ \omega_1 &= 0.5\left[\tanh\left(\beta(x_i - x_{i,MIN})\right) + 1\right] \\ \omega_2 &= 0.5\left[\tanh\left(\beta(x_i - x_{i,MAX})\right) + 1\right] \end{aligned} \tag{15}$$

Proof: The proof is completed by combining Lemma 6 and Eq. 8 together. □

Remark 4: Theorem 4 provides a continuous, smooth, and differentiable approach of approximation to evaluate self utility, reducing the accuracy requirements on both the hardware

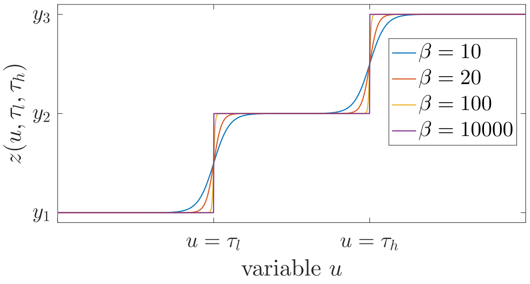

FIGURE 4. The shapes of $z(u, \tau_l, \tau_h)$ with varied β.

and software of each node greatly. It can be seen that when β is large enough, $z_i = \hat{f}_i$ holds perfectly. In this sense, algorithm convergence remains guaranteed by using z_i to replace $\hat{f}_i$. As a matter of fact, the β should be chosen as a constant that is several orders of magnitude larger than other variables. Also note that, a large β has little influence on the system performance because it is only an intrinsic parameter of the tanh function.

Remark 5: On the one hand, our algorithm is given in a continuous form and needs no time stepsize in fact. On the other hand, if necessary, our algorithm can be transformed to a discrete form by using proper discrete methods easily, because the algorithm is in a smooth form without any singular point. However, no matter which form it takes, our algorithm imposes a mild requirement on the stepsize factor (which is γ). To show this, we take $\theta-$ logarithmic barrier algorithm Eq. (18) as an example to illustrate the stepsize drawback of other algorithms. First of all, one can make such an observation that the stepsize factor θ in Eq. (18) cannot be a constant, because the feedback term $\Delta \triangleq \left(\frac{1}{x_i - x_{i,MIN}} - \frac{1}{x_{i,MAX} - x_i}\right)$ will become extremely large as x_i gets closer either to $x_{i,MIN}$ or to $x_{i,MAX}$, and a constant θ will make the optimization diverge. Secondly, θ needs to be elaboratedly designed to keep x_i strictly constrained, because the feedback term Δ is actually the derivative of $\ln(x_i - x_{i,MIN}) + \ln(x_{i,MAX} - x_i)$, which is the unconstrained reformulation by using Lagrange function. As a result, an overshot x_i makes $\ln(x_i - x_{i,MIN}) + \ln(x_{i,MAX} - x_i)$ go wrong, and makes its solving principle invalid. On the contrary, the feedback term of our algorithm is a linear combination of node states. To this end, our algorithm allows the stepsize factor γ to be a constant, and does not suffer from the overshot states.

Finally, as a summary, the proposed algorithm is concluded as Algorithm 1.

V. ILLUSTRATIVE EXAMPLES

To illustrate the effectiveness of the algorithm, some examples are presented in this part. In the simulations, the general gradient is used as the default gradient descent method.

A. EXAMPLE 1: SIMPLE LINEAR UTILITIES

Consider a most common allocation case, where $f_i(x_i)$ simply equals to x_i for each node, targeting the minimization of the consensus metric $V = \sum_{i,j \in \mathcal{N}} |x_i - x_j|$. The number of nodes is $n = 6$. The nodes networking is supposed to has a chain topology, i.e., 1–2–3–4–5–6, because this kind of topology is most challenging for testing a distributed algorithm. The constraints are limited as

$$\begin{aligned} x_D = \sum_{1 \le i \le 6} x_i = 41 \\ \{x_{i,MIN} \text{ for } i = 1:6\} = \{2, 1, 5, 2, 3, 3\} \\ \{x_{i,MAX} \text{ for } i = 1:6\} = \{10, 3, 8, 3, 5, 20\} \end{aligned} \tag{16}$$

Given an arbitrary initial state for each node, for instance, $\mathcal{X}(0) = \{10, 1, 6, 2, 4, 18\}$ with corresponding $V(0) = 222$, the global optimal assignment can be known obviously as $\mathcal{X}^* = \{10, 3, 8, 3, 5, 12\}$ with corresponding $V^* = 138$, by using an elaborately-designed global optimization algorithm (e.g., Particle Swarm Optimization [35]) from a global centralized perspective.

We then use Algorithm 1 to optimize $\mathcal{X}(0)$ in a fully distributed way and compare it with the global optimum. Let $\beta = 1e6$ to ensure an excellent approximation, use $x_{i,MIN} = x_{i,MIN} + \epsilon$ and $x_{i,MAX} = x_{i,MAX} - \epsilon$ to further ease the numerical computation with tolerance $\epsilon = 1e-6$ (small amount), and let $\gamma = 1$. The result is as follows. Trajectories of the states are shown in Fig. 5 and indicate that the optimization has convergence by using Algorithm 1. Trajectories of the errors between $\mathcal{X}$ and $\mathcal{X}^*$ are figured in Fig. 6, showing that the errors can converge to zero, i.e., Algorithm 1 can converge to the global optimum. The trajectory of the consensus metric V is figured in Fig. 7, which also indicates that the consensus metric can be optimized to the global optimum V^*. We should point out that this example

Algorithm 1 Distributed Constrained Consensus Algorithm

```
Input: $x_i(0)$, $x_{i,MIN}$, $x_{i,MAX}$, $f_i(x_i)$. Output: $x_i^*$
Each node $i \in \mathcal{N}$ behaves in the same manner as follows.
·At $t \le 0$: By communication once, obtains $\overline{f_{MAX}}$ and $\underline{f_{MIN}}$
·For any $t > 0$ (Our algorithm runs here):
    1) obtains $z_i$, following Eq. (15) with $\beta$ large enough.
    2) updates $x_i$, following Eq. (9) with $\hat{f}_i = z_i$.
·At $t = T$, with $T$ large enough: $x_i^* = x_i(T)$
```

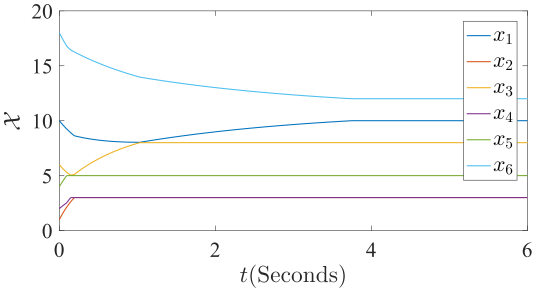

FIGURE 5. Trajectories of the states in Example 1.

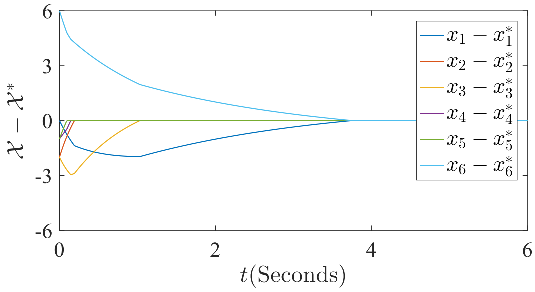

FIGURE 6. Trajectories of the errors between the states and corresponding benchmark optima in Example 1.

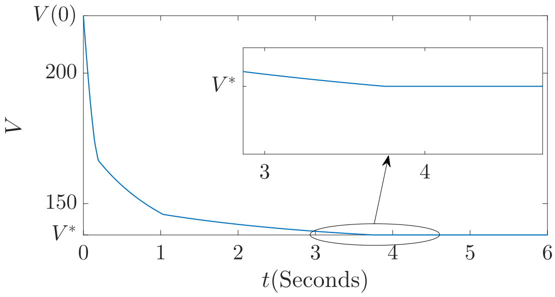

FIGURE 7. Trajectory of the consensus metric *V* in Example 1.

is complex enough for distributed optimization. Traditional unconstrained consensus algorithms would fail to solve it.

B. EXAMPLE 2: NONLINEAR UTILITIES

Consider a more complex allocation case, where $f_i(x_i)$ is a nonlinear function of x_i for each node, targeting the minimization of $V = \sum_{i,j\in\mathcal{N}} |f_i - f_j|$. The number of nodes is $n = 8$, and once again, suppose that the nodes networking has a chain topology, i.e., 1–2–3–4–5–6–7–8. The utilities and constraints are defined as

$$\begin{aligned} f_i &= x_i(c_i - x_i), \quad i = 1, 2, 3, 4 \\ f_i &= c_i x_i \ln x_i, \quad i = 5, 6, 7, 8 \\ \{c_i \text{ for } i = 1:8\} &= \{27, 25, 15, 18, 12, 10, 15, 25\} \\ x_D &= \sum_{1\le i\le 8} x_i = 42 \\ \{x_{i,MIN} \text{ for } i = 1:8\} &= \{6, 5, 2, 3, 4, 1, 2, 1\} \\ \{x_{i,MAX} \text{ for } i = 1:8\} &= \{11, 10, 7, 9, 8, 5, 6, 5\} \end{aligned} \tag{17}$$

Suppose that the initial states are given by $\mathcal{X}(0) = \{10, 6, 5, 3, 6, 4, 4, 4\}$. with corresponding $V(0) = 3139.27$. Likewise, let $\beta = 1\text{e}6$ and $\gamma = 1$ for Algorithm 1, and use $x_{i,MIN} = x_{i,MIN} + \epsilon$ and $x_{i,MAX} = x_{i,MAX} - \epsilon$, with $\epsilon = 1\text{e}{-6}$. We use Algorithm 1 for each node for 2 seconds, record the key variables at $t = 0$ and $t = 2$ in Table 1, and figure related trajectories in Fig. 8 to Fig. 11.

Trajectories of the states are plotted in Fig. 8, showing the convergence where the states converge to $\mathcal{X}(2) = \{6.00, 5.00, 7.00, 7.76, 4.44, 4.96, 3.90, 2.94\}$ with corresponding $V^* = 1185.4$. Note that nodes 1, 2 have arrived at their lower boundaries, node 3 has arrived at its upper boundary, and nodes 4, 5, 6, 7 lie within their open constraints. Trajectories of the utilities are figured in Fig. 9, from which one can see that each f_i converges individually but the converged utilities are not consistent for all nodes. This is because nodes {1, 2, 3} have reached their limits when attempting to coordinate their utilities consistent with the others. Fortunately, as a main feature of our algorithm, the utility evaluations do converge to consensus as Fig. 10 shows, where the violent fluctuation during $0 < t < 0.5$ indicates the frequent status-switching of nodes. Therefore, the full agreement of the utility evaluations guarantees that $\dot{x}_i = 0$ holds

TABLE 1. Results of example 2 at $t = 0$ and $t = 2$ by using Algorithm 1.

Node	1	2	3	4	5	6	7	8
$x_i(0)$	10.00	6.00	5.00	3.00	6.00	4.00	4.00	4.00
$x_i(2)$	6.00	5.00	7.00	7.76	4.44	4.96	3.90	2.94
$f_i(x_i(0))$	170.0	114.0	50.0	45.0	129.0	55.5	83.2	138.6
$f_i(x_i(2))$	126.0	100.0	56.0	79.5	79.5	79.5	79.5	79.5
$\hat{f}_i(x_i(0))$	170.0	114.0	50.0	45.0	129.0	55.5	83.2	138.6
$\hat{f}_i(x_i(2))$	79.5	79.5	79.5	79.5	79.5	79.5	79.5	79.5

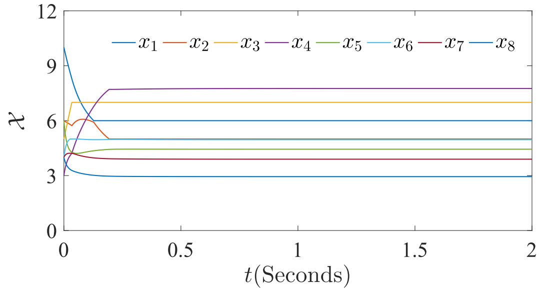

FIGURE 8. Trajectories of the states in Example 2.

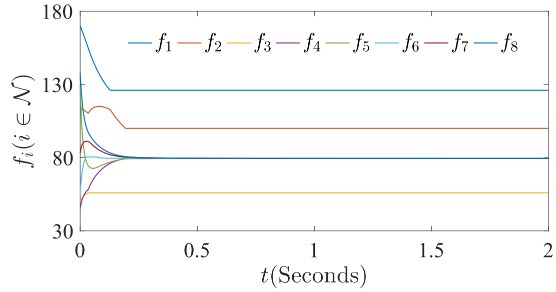

FIGURE 9. Trajectories of the utilities in Example 2.

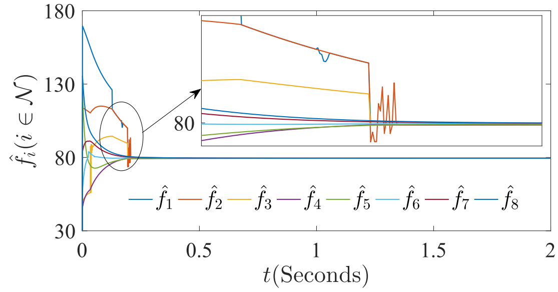

FIGURE 10. Trajectories of the utility evaluations in Example 2.

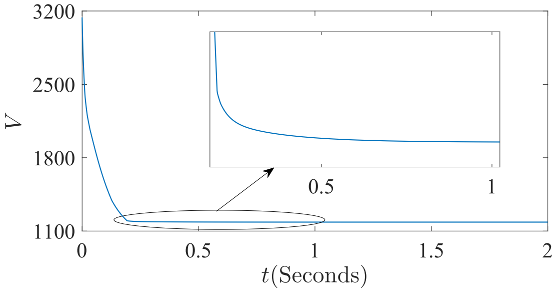

FIGURE 11. Trajectory of the consensus metric V in Example 2.

for all nodes at the stable equilibrium. Fig. 11 shows that V is decreasing with time and reaches the minimum. As analyzed in Theorem 1, it is clear that this equilibrium corresponds to the global minimization of V since any variation will cause V to increase. One can concludes finally that this example allocation problem has been optimized to the globally optimal assignment. It is worthwhile to mention that this example is complex enough for distributed optimization so that very few distributed algorithms can obtain the global optimum.

C. EXAMPLE 3: COMPARISON WITH $\theta-$ ALGORITHM

In this example, we compare Algorithm 1 with the benchmark algorithm, which refers to the $\theta-$ algorithm targeting the same problem in [31], to show the effectiveness and advantage of the former. For clarity, $\theta-$ algorithm is summarized here as Eq. (18), where the parameters of the benchmark are set to their best as given in [31].

$$\begin{aligned} \dot{x}_i &= -\sum_{j\in\mathcal{N}_i} a_{ij}(\phi_i - \phi_j) \\ \phi_i &= f_i - \frac{1}{\theta}\left(\frac{1}{x_i - x_{i,MIN}} - \frac{1}{x_{i,MAX} - x_i}\right) \\ \theta &\leftarrow 1.1\theta, \quad \text{when } x_i = x_{i,MIN} \,||\, x_i = x_{i,MAX} \end{aligned} \tag{18}$$

The same problem in Example 2 is used for testing the benchmark algorithm. The result converges to $\mathcal{X}^\theta = \{6.02, 5.05, 6.96, 7.83, 4.50, 4.79, 3.90, 2.95\}$ with corresponding $V^\theta = 1251.5$. Fig. 12 shows the trajectories of V by using Algorithm 1 and the benchmark respectively as a comparison. With $V^* = 1185.4$, it is obvious that V^θ is still larger than V^* by 5.6%, indicating the advantage of Algorithm 1. In fact, this is almost a best performance that the benchmark algorithm could achieve. As a matter of fact, the benchmark algorithm would even diverge in some cases. Although our algorithm can beat the benchmark with a much more enormous superiority by using more complex networks or setting different parameters, we would like to beat its best performance, and results show that this comparison is sufficient. The analysis of the failure of the benchmark can be made here once again. The reason lies exactly in the increasing variable θ. As mentioned previously, an enlarged θ in the benchmark algorithm will reduce the updating step and is harmful to global convergence. However, θ has to increase for avoiding divergence, overshooting, and algorithm errors. Further more, the benchmark algorithm will lose the ability to optimize a dynamical problem after θ gets large. On the contrary, our Algorithm 1 uses invariant parameters independent of time. This feature guarantees the invariant effectiveness of the algorithm. More similar examples can be performed likewise, but the conclusions are consistent and therefore we omit them for short.

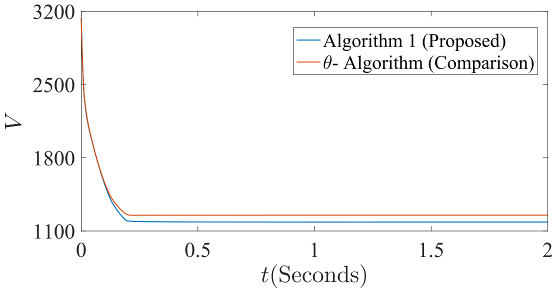

FIGURE 12. Trajectories of the consensus metric V by using Algorithm 1 and $\theta-$ algorithm, respectively.

VI. CONCLUSION

This paper investigates the problem of distributed constrained consensus of utilities for multiple networked nodes. Towards this goal, a generalized framework is developed via a self evaluation approach, in which each node uses individual evaluation to interact with its neighbors and drives its state to global optimization. This proposed framework is flexible for helping solve a class of general state-constrained consensus problems in a fully distributed way. A distributed algorithm is then built on this framework to solve utility consensus problem with constraints of states. Theoretical analyses are presented showing the convergence and stability. In addition, an approximation method is proposed for ease of engineering implementation as well. Finally, three illustrative examples are provided and analyzed to substantiate the efficacy of the proposed algorithm.

REFERENCES

[1] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[2] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.

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

[4] I. Matei and J. S. Baras, "Performance evaluation of the consensus-based distributed subgradient method under random communication topologies," *IEEE J. Sel.Topics Signal Process.*, vol. 5, no. 4, pp. 754–771, Aug. 2011.

[5] S. Lee and A. Nedic, "Distributed random projection algorithm for convex optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 221–229, Apr. 2013.

[6] D. Yuan, D. W. C. Ho, and S. Xu, "Regularized primal–dual subgradient method for distributed constrained optimization," *IEEE Trans. Cybern.*, vol. 46, no. 9, pp. 2109–2118, Sep. 2016.

[7] X. Wang, T. Sun, and C. Fan, "Neighborhood Kalman estimation for distributed localization in wireless sensor networks," *Math. Problems Eng.*, vol. 2016, Jan. 2016, Art. no. 2510903.

[8] G. Obando, N. Quijano, and N. Rakoto-Ravalontsalama, "A centre–free approach for resource allocation with lower bounds," *Int. J. Control*, vol. 90, no. 9, pp. 1830–1845, 2017.

[9] T. Yang, J. Lu, D. Wu, J. Wu, G. Shi, Z. Meng, and K. H. Johansson, "A distributed algorithm for economic dispatch over time-varying directed networks with delays," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 5095–5106, Jun. 2017.

[10] Z. Meng, T. Yang, G. Li, W. Ren, and D. Wu, "Synchronization of coupled dynamical systems: Tolerance to weak connectivity and arbitrarily bounded time-varying delays," *IEEE Trans. Autom. Control*, vol. 63, no. 6, pp. 1791–1797, Jun. 2018.

[11] X. He, J. Yu, T. Huang, and C. Li, "Distributed power management for dynamic economic dispatch in the multimicrogrids environment," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 4, pp. 1651–1658, Jul. 2019.

[12] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Syst. Mag.*, vol. 27, no. 2, pp. 71–82, Apr. 2007.

[13] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 863–868, May 2007.

[14] M. Patriksson, "A survey on the continuous nonlinear resource allocation problem," *Eur. J. Oper. Res.*, vol. 185, no. 1, pp. 1–46, Feb. 2008.

[15] T. Li, M. Fu, L. Xie, and J.-F. Zhang, "Distributed consensus with limited communication data rate," *IEEE Trans. Autom. Control*, vol. 56, no. 2, pp. 279–292, Feb. 2011.

[16] J. He, P. Cheng, L. Shi, J. Chen, and Y. Sun, "Time synchronization in WSNs: A maximum-value-based consensus approach," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 660–675, Mar. 2014.

[17] T. Yang, Z. Meng, G. Shi, Y. Hong, and K. H. Johansson, "Network synchronization with nonlinear dynamics and switching interactions," *IEEE Trans. Autom. Control*, vol. 61, no. 10, pp. 3103–3108, Oct. 2016.

[18] Z. Meng, T. Yang, G. Shi, D. V. Dimarogonas, Y. Hong, and K. H. Johansson, "Targeted agreement of multiple lagrangian systems," *Automatica*, vol. 84, pp. 109–116, Oct. 2017.

[19] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *J. Optim.*, vol. 25, no. 2, pp. 944–966, May 2015.

[20] Z. Qiu, S. Liu, and L. Xie, "Distributed constrained optimal consensus of multi-agent systems," *Automatica*, vol. 68, pp. 209–215, Jun. 2016.

[21] P. Xie, K. You, R. Tempo, S. Song, and C. Wu, "Distributed convex optimization with inequality constraints over time-varying unbalanced digraphs," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4331–4337, Dec. 2018.

[22] A. Nedić, A. Olshevsky, and W. Shi, "Decentralized consensus optimization and resource allocation," in *Large-Scale and Distributed Optimization*. Cham, Switzerland: Springer, 2018, pp. 247–287.

[23] X. Wang, S. Mou, and B. D. O. Anderson, "Scalable, distributed algorithms for solving linear equations via double-layered networks," *IEEE Trans. Autom. Control*, to be published.

[24] D. Yuan and D. W. C. Ho, "Randomized gradient-free method for multiagent optimization over time-varying networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1342–1347, Jun. 2015.

[25] A. Nedić and A. Olshevsky, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 3936–3947, Dec. 2016.

[26] Q. Liu, S. Yang, and J. Wang, "A collective neurodynamic approach to distributed constrained optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 8, pp. 1747–1758, Aug. 2017.

[27] G. Shi, K. H. Johansson, and Y. Hong, "Reaching an optimal consensus: Dynamical systems that compute intersections of convex sets," *IEEE Trans. Autom. Control*, vol. 58, no. 3, pp. 610–622, Mar. 2013.

[28] W. Chen and W. Ren, "Event-triggered zero-gradient-sum distributed consensus optimization over directed networks," *Automatica*, vol. 65, pp. 90–97, Mar. 2016.

[29] A. Wang, X. Liao, and T. Dong, "Event-triggered gradient-based distributed optimisation for multi-agent systems with state consensus constraint," *IET Control Theory Appl.*, vol. 12, no. 10, pp. 1515–1519, Mar. 2018.

[30] C. Sun, X. Wang, and J. Liu, "Evolutionary game theoretic approach for optimal resource allocation in multi-agent systems," in *Proc. Chin. Autom. Congr. (CAC)*, Oct. 2017, pp. 5588–5592.

[31] C. Li, X. Yu, T. Huang, and X. He, "Distributed optimal consensus over resource allocation network and its application to dynamical economic dispatch," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2407–2418, Jun. 2018.

[32] W. Xiaochu, S. Changhao, and S. Ting, "A novel distributed algorithm for constrained resource allocation," in *Proc. 11th Int. Conf. Intell. Hum.-Mach. Syst. Cybern.*, to be published.

[33] Y. Cao and W. Ren, "Distributed coordinated tracking with reduced interaction via a variable structure approach," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 33–48, Jan. 2012.

[34] A. F. Filippov, *Differential Equations with Discontinuous Righthand Sides*. Amsterdam: The Netherlands: Kluwer, 1988.

[35] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.

XIAOCHU WANG received the B.S. degree in micro electro-mechanical system engineering and the Ph.D. degree in instrument science and technology from Tsinghua University, Beijing, China, in 2009 and 2014, respectively. Since 2014, he has been a Research Associate with the Qian Xuesen Laboratory of Space Technology, China Academy of Space Technology, Beijing. His research interests include spacecraft navigation, attitude estimation and control, and multi-agent systems. His current research interests include distributed estimation, optimization, and control in multi-agent systems.

CHANGHAO SUN received the B.S. degree in automation from the Beijing Institute of Technology, Beijing, China, in 2010, and the Ph.D. degree in guidance, navigation, and control from Beihang University, Beijing, in 2016. Since 2016, he has been a Research Associate with the Qian Xuesen Laboratory of Space Technology, China Academy of Space Technology, Beijing. His main research interests include swarm intelligence computation and (evolutionary) game theory with applications to distributed coordination, and optimization in multi-agent systems.

TING SUN received the B.S. degree in measurement and control technology and instruments and the Ph.D. degree in instrument science and technology from Tianjin University, Tianjin, China, in 2009 and 2014, respectively. Since 2014, she has been a Research Associate with the School of Photoelectronic Information and Communication Engineering, Beijing Information Science and Technology University, Beijing, China. Her main research interests include instrument technology, aerospace science, information fusion, and optimization algorithms.