Distributed delay-tolerant strategies for equality-constraint sum-preserving resource allocation

Mohammadreza Doostmohammadian [a,*], Alireza Aghasi [b], Maria Vrakopoulou [c], Hamid R. Rabiee [d], Usman A. Khan [e], Themistoklis Charalambous [f,g]

[a] *Faculty of Mechanical Engineering, Semnan University, Semnan, Iran*
[b] *Department of Electrical Engineering and Computer Science, Oregon State University, USA*
[c] *University of Melbourne, Melbourne, Australia*
[d] *Department of Computer Engineering, Sharif University of Technology, Tehran, Iran*
[e] *Electrical and Computer Engineering Department, Tufts University, MA, USA*
[f] *School of Electrical Engineering, Aalto University, Espoo, Finland*
[g] *School of Electrical Engineering, University of Cyprus, Nicosia, Cyprus*

ARTICLE INFO

ABSTRACT

This paper proposes two nonlinear dynamics to solve constrained distributed optimization problem for resource allocation over a multi-agent network. In this setup, coupling constraint refers to resource-demand balance which is preserved at all-times. The proposed solutions can address various model nonlinearities, for example, due to quantization and/or saturation. Further, it allows to reach faster convergence or to robustify the solution against impulsive noise or uncertainties. We prove convergence over weakly connected networks using convex analysis and Lyapunov theory. Our findings show that convergence can be reached for general sign-preserving odd nonlinearity. We further propose delay-tolerant mechanisms to handle general bounded heterogeneous time-varying delays over the communication network of agents while preserving all-time feasibility. This work finds application in CPU scheduling and coverage control among others. This paper advances the state-of-the-art by addressing *(i)* possible nonlinearity on the agents/links, meanwhile handling *(ii)* resource-demand feasibility at all times, *(iii)* uniform-connectivity instead of all-time connectivity, and *(iv)* possible heterogeneous and time-varying delays. To our best knowledge, no existing work addresses contributions *(i)-(iv)* altogether. Simulations and comparative analysis are provided to corroborate our contributions.

1. Introduction

1.1. Background

Distributed algorithms have gained considerable attention because of recent advances in the Internet of Things (IoT), cloud computing, and parallel processing. Distributed optimization over multi-agent networks, in particular, has emerged as an effective solution in large-scale applications ranging from machine-learning for large-scale data mining [1,2] to power flow optimization over the smart grid [3–6]. Distributed/decentralized algorithms outperform centralized ones in many aspects: no-single-node-of-failure, scalability, speed/efficiency,

etc. Distributed or decentralized optimization allows for scaling the problem-solving process to handle larger and more complex optimization problems by distributing the computational load across multiple nodes. If one of the node fails, the others can still solve the remaining objective function and this makes distributed optimization inherently more resilient to node-failures. Distributed resource allocation can use parallel processing to speed up the optimization process. By dividing the problem into sub-problems and solving them concurrently, the overall solution can be obtained faster and computationally more efficient than a centralized approach. In this work, we propose a distributed setup for resource allocation over general multi-agent systems with constraints on the communication network or the agents.

---

* Corresponding author.
E-mail addresses: doost@semnan.ac.ir (M. Doostmohammadian), alireza.aghasi@oregonstate.edu (A. Aghasi), maria.vrakopoulou@unimelb.edu.au (M. Vrakopoulou), rabiee@sharif.edu (H.R. Rabiee), khan@ece.tufts.edu (U.A. Khan), themistoklis.charalambous@aalto.fi, charalambous.themistoklis@ucy.ac.cy (T. Charalambous).

1.2. Challenges

In real-world applications, many model nonlinearities exist in practice, some of which stem from the inherent physical constraints, e.g., actuator saturation or quantized data exchange [7,8], and some are purposely added, e.g., to improve the convergence rate [9,10] or robustness to impulsive noise and uncertainties [11]. For example, in automatic generation control setup for power systems there exist ramp-rate-limit (RRL) due to limited rate of power generation by the real-world generators. In other words, the increase and decrease of the power generation cannot follow any rate but is limited. Most existing results in the literature cannot address this limit and therefore the generators cannot follow the rates assigned by these solutions. Another example is when the information exchange among the agents is quantized. Most existing linear solutions do not address quantized information exchange among the agents and, in this aspect, the algorithm is not realistic. Such unseen nonlinearities make networked optimization (both constrained and unconstrained) more challenging in terms of computation, accuracy, feasibility, optimality, and convergence. Further, the network itself might be subject to time delays, or asynchronous data-transmission [5,12–15]. The network also may lose network connectivity over some intermittent time-intervals (e.g., due to packet loss or link failure). The notion of constraint feasibility is another challenging issue. The equality-constraint ensures the resource-demand balance, and its violation may cause service disruption. This paper aims to address general nonlinear solutions while handling network latency, uniform network connectivity, and all-time feasibility. The objective function is also nonlinear and this nonlinearity might be due to addressing the box constraints as penalty terms.

1.3. Literature review

The literature on distributed optimization, both constrained and unconstrained, mainly assumes linear models and no delay for data exchange over the network. There exist a few nonlinear reinforcement-learning-based models for resource allocation and economic dispatch [16,17]. These works find optimal allocation with no prior knowledge of the mathematical formulation of the actual generation costs. In [16] a Neural-Network is used to learn the relationship between the demand and the optimal output of each generation unit, while [17] combines the state–action-value function approximation with a distributed optimization based on multiplier splitting. Some existing works focus on one specific nonlinearity, e.g., quantized allocation/information-sharing [7,18–23] or saturated dynamics [7,24]. Some others are devoted to address sign-based nonlinearities to improve the convergence rate, e.g., to reach the optimal value in finite-time [25,26] or fixed-time [9,10,27]. Such fixed/finite dynamics are also prevalent in consensus literature [11,25,28] which also allow for robust and noise-resilient design. However, these solutions are designed for a *specific* case and cannot handle *composition* of two or more nonlinearities. Moreover, in many applications, the network is dynamic (time-varying) due to, e.g., agents' mobility and may even get disconnected over some time intervals. Therefore, it is more practical to assume *uniform connectivity*, in contrast to all-time connectivity in many existing solutions [29,30]. Latency is another networking issue that may cause the optimization algorithm to diverge. Few works in this literature address possible *homogeneous* delays at all links or asynchronous communication [15,31], with no consideration of model nonlinearities. In the sum-preserving constrained optimization, another challenge (other than stability) is to preserve all-time feasibility [6,29] (in contrast to asymptotic feasibility [15,32,33]). All-time feasibility implies that, as the solution evolves, the constraint on the states always holds. For example, in the economic dispatch problem (EDP), at any termination point of the algorithm, the sum of the power states must be equal to the load demand. Otherwise, it causes disruption, power delivery issues, and even system breakdown [6,29]. Such all-time feasibility conditions

cannot be addressed by dual-based solutions, e.g., alternating-direction-method-of-multipliers (ADMM) methods [34–40]. These works claim to reach feasibility fast enough within the running interval of the algorithm. Besides, the mentioned nonlinearities, network variation, uniform connectivity, and latency have not been addressed by the existing ADMM solutions [34–39]. Some of the existing algorithms, on the other hand, solve specific quadratic-form problems, for example, consensus-based solutions for CPU scheduling [30] or economic dispatch [41]. In general, however, the objective could be non-quadratic, e.g., because of additional penalty terms and barrier functions to handle the so-called *box constraints*.

1.4. Motivations

What is missing in the existing literature is a general method to address "model nonlinearitiy" to solve (general) non-quadratic objectives while considering uniform connectivity, latency, and sum-preserving all-time feasibility altogether. The nonlinearity of the node dynamics might be due to, for example, ramp-rate-limit (RRL) of the generators in automatic generation control (AGC) [42]. The other common issue is quantized information exchange in real-world communication networks that motivates the use of algorithm which can handle such a nonlinearity. In this work, we kept the application general and we focus on general multi-agent systems with communication networks. The communication networks often introduce delays due to packet processing, transmission time, and network congestion. Additionally, agents are subject to some processing time before state-update or message-sharing. Therefore, the information sent from one agent/node may reach the receiver agent with certain time-delay. These delays may cause divergence of the solution and/or feasibility and optimality gap. This motivates the need to introduce a delay-tolerant solution for real-world applications. Furthermore, the network itself might be subject to some changes and lose connectivity over intermittent time-intervals due to, for example, agents' mobility. This motivates the use of algorithms to handle uniform-connectivity instead of all-time network connectivity. Another motivation is to address all-time feasibility of the equality-constraint to ensure resource-demand balance at all times. This guarantees that there is no service disruption before the termination of the algorithm at any time. This work finds many applications, e.g., in CPU scheduling [43], coverage control [44], and plug-in electric vehicle (PEV) charging coordination [45] among others.

1.5. Main contributions

In this work, we propose two general nonlinear solutions for distributed constrained convex optimization: one addressing the nonlinearity of the nodes and one of the links. Sufficient conditions on the nonlinearity and network connectivity to ensure convergence are discussed. The Lyapunov-type proof analysis is irrespective of the specific type of nonlinearity. Therefore, the algorithm can address certain inherent physical model nonlinearity (e.g., quantization and/or saturation) or purposely added nonlinearity (e.g., signum-based) to *design* fast-convergent or robust-to-noise solutions. Our model can handle the composition of more than one nonlinear mapping. In power networks, for example, the saturated generator dynamics due to RRLs is not addressed by the existing methods [4,6,26,31,41,46,47] in the AGC setup. Our proposed general multi-agent network can be adapted to communicate and transmit quantized information, address actuator saturation, reach a tunable (or predefined) rate of convergence, embrace resiliency and impulsive-noise-tolerance, or any composition of such nonlinear models. We prove coupling-constraint feasibility and convergence subject to general upper/lower sector-bound nonlinearity. In other words, sufficient conditions on the model's nonlinearity are derived to not violate all-time sum-preserving feasibility (resource-demand balance) and convergence to the optimal value; see examples in Section 8.

We discuss both continuous-time (CT) and discrete-time (DT) solutions. In the DT case, we design delay-tolerant solutions to handle finite *arbitrary* and *heterogeneous* time delays over the network. We prove constraint feasibility and convergence/stability under bounded step sizes and certain assumptions on the time-delays. Two approaches to handle network latency are given: (i) Case I, updating over a longer time scale after receiving all delayed information, and (ii) Case II, updating by all the information received at the same time scale as the DT dynamics. The proposed delay-tolerant solutions lead to *no feasibility gap* over switching and uniformly connected networks (instead of all-time connectivity) and in the presence of heterogeneous delays and model nonlinearity. This is in contrast to, e.g., the solution by [15] with some feasibility gap. The uniform connectivity is motivated by applications in mobile sensor networks, where the links may come and go as the mobile sensors (or robots) move into and out of line-of-sight (or broadcast range) of each other. The network may sometimes even lose connectivity due to link failure or even packet loss, while it maintains *uniform connectivity* over some finite time intervals. Finally, we apply our proposed solutions in distributed setup with *(i)* quadratic costs and *(ii)* non-quadratic costs with logarithmically quantized values.

We summarized our contributions in the following:

(i) Possible inherent and additive nonlinearity in the model dynamics can be addressed by our proposed resource allocation model with general non-quadratic objective functions allowing for consideration of penalty/barrier functions.

(ii) Our proposed delay-tolerant solution can handle possible heterogeneous and arbitrary delays of the links over general uniformly-connected networks. The values of delays might be different at the links and change over time while the network may lose connectivity over some intermittent time-intervals.

(iii) Our proposed distributed resource allocation is all-time feasible implying that at all times the resource-demand balance holds even in the presence of time-delays and losing network connectivity. This is in contrast to dual formulation methods in which the feasibility gap asymptotically reduces over time. The all-time resource-demand feasibility prevents service disruption at any termination/assignment time of the algorithm.

To the best of our knowledge, no previous works in the literature address contributions *(i)-(iii)* altogether.

1.6. Paper organization

We introduce the preliminary notions and some useful lemmas to set up the problem in Section 2. The CT solutions are proposed in Section 3 and their convergence is discussed in Section 4. The DT counterparts subject to time delays are discussed in Sections 5 and 6. Applications and more simulations on sparse dynamic networks are presented in Sections 7 and 8. Finally, Section 9 concludes the paper.

1.7. General notation

In this paper, we present the column vectors in bold small letters and scalars with small letters. The matrices are denoted by capital letters. For notation simplicity, ∂f_i and $\partial^2 f_i$ denote $\frac{df_i(x_i)}{dx_i}$ and $\frac{d^2 f_i(x_i)}{dx_i^2}$, respectively. See the full list of notations in Table 1.

The communication network of agents (also referred to as the multi-agent network) considered in this paper is modelled by a (possibly) time-varying graph $\mathcal{G}(t) = \{\mathcal{V}, \mathcal{E}(t)\}$ with a set of time-dependent links $\mathcal{E}(t)$ and time-invariant set of nodes $\mathcal{V} = \{1, \ldots, n\}$. A link $(i,j) \in \mathcal{E}(t)$ represents possible information exchange (communication) from the agent i to the agent j, and further implies that the agent i is in the neighbouring set of the agent j, defined as $\mathcal{N}_j(t) = \{i | (i,j) \in \mathcal{E}(t)\}$. The link $(j,i) \in \mathcal{E}(t)$ is weighted by $W_{ij} > 0$ and $W(t) = [W_{ij}(t)] \in \mathbb{R}_{\geq 0}^{n \times n}$ represents the weight matrix of the network $\mathcal{G}(t)$. Clearly, $W(t)$ follows

Table 1
List of notations.

x, z	Column state vectors
$\mathbf{1}_n, \mathbf{0}_n$	column vector of ones and zeros of size n
$\mathbf{x}^*, \mathbf{z}^*$	optimal state vector
$\widetilde{S}_b, S_b$	feasibility sets (associated with b) of $\mathbf{z}$ and $\mathbf{x}$ states
x_i, z_i	element i of the state vector (state of node i)
t, k	continuous and discrete time index
$F(\cdot)$	global cost function
$\mathcal{G}, \mathcal{V}, \mathcal{E}$	network, set of nodes, set of links
$\nabla F(\cdot)$	gradient of $F(\cdot)$
d_g	network diameter
$f_i(\cdot)$	local cost function
W, L	weight matrix, Laplacian matrix
$\partial f_i(\cdot)$	first derivative of $f_i(\cdot)$
λ_i	ith eigenvalue of L
$\partial^2 f_i(\cdot)$	second derivative of $f_i(\cdot)$
τ_{ij}	time-delay of link (j,i) (or $j \to i$)
$\mathcal{N}_i$	set of neighbours of node i
$\overline{\tau}$	global upper-bound on the time-delay
K_f	Lipschitz constant of $f(\cdot)$
T	sampling step
$g(\cdot)$	nonlinear mapping
$\mathcal{I}(\cdot)$	indicator function
n	number of agents/nodes
$|\cdot|, \lceil \cdot \rceil, \lfloor \cdot \rfloor$	absolute value, ceil function, floor function

the structure (zero-nonzero pattern) of the adjacency matrix associated with $\mathcal{G}(t)$. In $\mathcal{G}(t)$, we define a *spanning tree* as a sub-graph of size n (covering all nodes) in which there is only one path between every two nodes [48]. This spanning tree is known to include minimum possible links for connectivity [48]. Define the Laplacian matrix $L(t) = [L_{ij}(t)]$ as,

$$L_{ij}(t) = \begin{cases} \sum_{j \in \mathcal{N}_i} W_{ij}(t), & \text{for } i = j, \\ -W_{ij}(t), & \text{for } i \neq j. \end{cases} \tag{1}$$

2. Problem setup

This paper considers equality-constraint optimization problems in the primal nonlinear formulation over a multi-agent network. The objective of the problem is to minimize the cost function while satisfying the (weighted) resource-demand balance constraint. This balance equality-constraint ensures that the weighted sum of resources meets the demand by the user, otherwise it may cause service disruption in the application under consideration. The mathematical formulation is as follows:

$$\min_{\mathbf{z}} \quad \widetilde{F}(\mathbf{z}) := \sum_{i=1}^n \widetilde{f}_i(z_i) \tag{2}$$

$$\text{s.t.} \quad \mathbf{z}^\top \mathbf{a} = b$$

where element $z_i \in \mathbb{R}$ represents the state variable at agent i, Column vector $\mathbf{z} = [z_1; \ldots; z_n] \in \mathbb{R}^n$ is the collective vector state,[1] and column vector $\mathbf{a} = [a_1; \ldots; a_n] \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are the constraint parameters. $f_i : \mathbb{R} \mapsto \mathbb{R}$ is the local cost (or loss) function at agent i, and the overall cost function is $F : \mathbb{R}^n \mapsto \mathbb{R}$. The problem can be also extended to consider box constraints on the states, i.e., $\underline{z}_i \leq z_i \leq \overline{z}_i$. In this case, one can eliminate these extra constraints by adding proper exact (nonlinear) penalty functions to every local objective; for example, changing the local objectives to $f_i^\sigma(z_i) = f_i(z_i) + c([z_i - \overline{z}_i]^+ + [\underline{z}_i - z_i]^+)$, with $[u]^+ = \max\{u, 0\}^\sigma$, $\sigma \in \mathbb{N}$, and $c \in \mathbb{R}^+$. Some other example penalties and barrier functions are discussed in [51]. In general, such penalty (or barrier) functions are non-quadratic and nonlinear, which makes the objective function non-quadratic and nonlinear; see examples in [29,

---

[1] In its most general form, the problem can be extended to $\mathbf{z}_i \in \mathbb{R}^m$ with $m > 1$ (as in [49,50]). Considering $m = 1$ is for the sake of simplifying the proof analysis in Sections 4 and 5.

49]. *Feasible initialization* algorithms under such local constraints are discussed, for example, in [6].

The problem is generally stated in the following standard sum-preserving form

$$
\min_{\mathbf{x}} \quad F(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i)
$$
$$
\text{s.t.} \quad \sum_{i=1}^{n} x_i = b \tag{3}
$$

which can be obtained from (2) by simple change of variables $z_i a_i =: x_i$, with box constraints transformed into $a_i \underline{z}_i \leq x_i \leq a_i \overline{z}_i$ for $a_i > 0$ and reversed otherwise. We aim to provide general nonlinear dynamics to solve problem (3) over a multi-agent network for different applications. The proposed solutions need to be distributed, implying that the information available at each agent i includes its own information (for example, its state and local objective) and the data received from agents $j \in \mathcal{N}_i$ (its direct neighbours). This work addresses the case that agents/links are constrained with some nonlinearity. Moreover, the proposed distributed solution remains feasible (i.e., $\sum_{i=1}^{n} x_i(t) = b$, $\forall t \geq 0$) and delay-tolerant (under heterogeneous time-delays). Further, it is possible that the communication network changes over time and loses connectivity at some bounded time-intervals. The assumptions on the objective function convexity (to include possible penalty terms), the network connectivity, feasibility, and the time-delay model are essential in the problem setup, as discussed next.

2.1. Useful lemmas and definitions on convexity

Definition 1 (*Lipschitz Condition*). Let $h : \mathbb{R} \mapsto \mathbb{R}$ represent a nonlinear mapping. Function $h(y)$ is called Lipschitz continuous if there exists a real constant K_h such that for any $y_1, y_2 \in \mathbb{R}$,

$$
|h(y_1) - h(y_2)| \leq K_h |y_1 - y_2|. \tag{4}
$$

Definition 2 (*Strict Convexity*). A function $f : \mathbb{R} \mapsto \mathbb{R}$ is strictly convex if $\forall y_1, y_2 \in \mathbb{R}, \ \forall \kappa \in (0, 1)$,

$$
f(\kappa y_1 + (1 - \kappa)y_2) < \kappa f(y_1) + (1 - \kappa)f(y_2). \tag{5}
$$

It is known that for a smooth strictly convex function, $\partial^2 f(y) > 0$ for $y \in \mathbb{R}$ [52].

Assumption 1. The local cost/objective functions $f_i(x_i) : \mathbb{R} \mapsto \mathbb{R}$, $i \in \{1, \dots, n\}$ are smooth and strictly convex, i.e., $\partial^2 f_i(x_i) > 0$.

Note that the penalized objective function f_i^{σ} is also Lipschitz since $[u]^+$ is Lipschitz for $\sigma \in \mathbb{N}_{\geq 2}$. Recalling the Taylor series expansion, the following holds.

Lemma 1 (*[46]*). *Given a continuous strictly-convex function $f(y)$, two points y_1, y_2, and $\Delta y =: y_1 - y_2$, there exist $\overline{y} := \kappa y_1 + (1 - \kappa)y_2, 0 < \kappa < 1$ such that,*

$$
f(y_1) = f(y_2) + \nabla F(y_2)^\top \Delta y + \frac{1}{2} \Delta y^\top \partial^2 f(\overline{y}) \Delta y. \tag{6}
$$

Lemma 2 (*[52]*). *For cost function $f_i : \mathbb{R} \mapsto \mathbb{R}$, assume $2v < \partial^2 f_i < 2u$ with $0 < v \leq u < \infty$.[2] Then, for two points $x_1, x_2 \in \mathbb{R}$, and $\Delta x := x_1 - x_2$, the following statements hold:*

$$
F(x_1) < F(x_2) + \nabla F(x_1)\Delta x + u \Delta x \Delta x. \tag{7}
$$
$$
F(x_1) > F(x_2) + \nabla F(x_1)\Delta x + v \Delta x \Delta x.
$$

Lemma 3. *Let Assumption 1 hold. The unique optimal solution $\mathbf{x}^* = [x_1^*; \dots; x_n^*]$ to problem (3) is in the form,*

$$
\nabla F(\mathbf{x}^*) = \varphi^* \mathbf{1}_n, \tag{8}
$$

with $\mathbf{1}_n$ as the column vector of 1s, $\partial f_i \in \mathbb{R}$ as the gradient of the local function $f_i(\cdot)$, $\nabla F(\mathbf{x}^*) = [\partial f_1(\mathbf{x}_1^*); \dots; \partial f_n(\mathbf{x}_n^*)] \in \mathbb{R}^n$, and $\varphi^* \in \mathbb{R}$.

Proof. The proof follows from the KKT condition and the Lagrange multipliers method [46,52]. □

Similarly, the unique solution of problem (2) is in the form $\nabla \widetilde{F}(\mathbf{z}^*) = \widetilde{\varphi}^* \mathbf{a}$. Note that, in the presence of the box constraints, the above lemma assumes that $\mathbf{z}^*$ meets those constraints, i.e., $\underline{z}_i \leq z_i^* \leq \overline{z}_i$ for all i.

Throughout the paper, we refer to the feasibility condition as described below.

Definition 3 (*Feasibility*). Define the feasible sets $\widetilde{S}_b = \{\mathbf{z} \in \mathbb{R}^n | \mathbf{z}^\top \mathbf{a} = b\}$ in (2) or $S_b = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{x}^\top \mathbf{1}_n = b\}$ in (3). A feasible solution then is defined as $\mathbf{x} \in S_b$ or $\mathbf{z} \in \widetilde{S}_b$.

Lemma 4. *Let Assumption 1 hold. Initializing from any feasible set $\widetilde{S}_b$ there is only one unique point $\mathbf{z}^* \in \widetilde{S}_b$ such that $\nabla \widetilde{F}(\mathbf{z}^*) = \widetilde{\varphi}^* \mathbf{a}$ for $\widetilde{\varphi}^* \in \mathbb{R}$. Similarly, there is unique $\mathbf{x}^* \in S_b$ such that $\nabla F(\mathbf{x}^*) = \varphi^* \mathbf{1}_n$.*

Proof. The proof follows from the strict convexity of the function $F(\mathbf{x})$ (and $\widetilde{F}(\mathbf{x})$) based on Assumption 1. For detailed proof based on level-set analysis see [18,49]. □

2.2. Recall on graph theory

Assumption 2. The following assumptions hold.[3]

(a) Every link in $\mathcal{G}(t)$ is bidirectional with the same weight on both sides at all times. This implies that the weight matrix $W(t)$ is symmetric and balanced for $t \geq 0$.

(b) $\mathcal{G}(t)$ is *uniformly connected* over time-window $B > 0$ (or B-connected), implying that there exists $B > 0$ such that the (edge) union graph $\mathcal{G}_B(t) = \{\mathcal{V}, \mathcal{E}_B(t)\}$ includes a spanning tree for every $t \geq 0$ where,

$$
\mathcal{E}_B(t) = \bigcup_{t}^{t+B} \mathcal{E}(t).
$$

The bidirectional condition in Assumption 2 holds, for example, when agents/nodes have similar broadcasting levels and their communication range is the same. Therefore, if $i \in \mathcal{N}_j(t)$ then $j \in \mathcal{N}_i(t)$ while the assigned weights are the same $W_{ij}(t) = W_{ji}(t)$ at all time. Later in the paper, this assumption is extended to weight-balanced directed networks with $\sum_{i=1}^{n} W_{ij}(t) = \sum_{j=1}^{n} W_{ij}(t)$ for some particular cases. This B-connectivity condition in Assumption 2 is considerably weaker than all-time connectivity in many works (e.g., the ADMM solutions [34–40]). In particular, this allows $\mathcal{G}$ to lose connectivity over some time intervals. In other words, the connectivity requirement needs to be satisfied over longer time intervals in the case of links arbitrarily coming and going over the dynamic network. Such an assumption is known to be the least-connectivity assumption in consensus and distributed optimization literature [53,54]. An example contradicting this condition is the case when $\mathcal{G}(t)$ contains two separate sub-graphs $\mathcal{G}_1(t)$ and $\mathcal{G}_2(t)$ with no path between them for $t > t_0$, implying that no consensus can be achieved between the two. Recall that the sparse connectivity in Assumption 2(b) is strong enough for our algorithm to ensure convergence over the network (as shown later).

---

[2] The condition $2v < \partial^2 f_i(x_i)$ implies that the cost function f_i is strongly convex, see *Assumption 5* in Section 5.1.

[3] In this paper, we generally assume that the network is undirected for the delay-tolerant case. Extension to *balanced directed graphs* is discussed later in Remark 5.

Remark 1. The network conditions in Assumption 2 are less restrictive than the existing literature. In particular, the weight-balanced and symmetric condition (a) relaxes the weight-stochasticity in [30,35], while uniform-connectivity (b) relaxes all-time connectivity in [6,29, 34,46]. These further motivate analysis of *dynamic* networks under link removal and packet loss, with no need to re-adjust (update) the link weights for ensuring stochasticity [55].

The next lemma gives an intuition to relate the dispersion of the entries in **x** with the eigen-spectrum of L. This mainly follows from the Courant–Fischer theorem and, for example, gives an estimate on the *disagreement* value in consensus algorithms [56].

Lemma 5. *Consider a symmetric Laplacian matrix L (of a graph $\mathcal{G}$) and vector $\mathbf{x} \in \mathbb{R}^n$. Define the so-called dispersion state vector as $\bar{\mathbf{x}} =: \mathbf{x} - \frac{\mathbf{1}_n^\top \mathbf{x}}{n} \mathbf{1}_n$. The following statements hold:*

$$\mathbf{x}^\top L \mathbf{x} = \bar{\mathbf{x}}^\top L \bar{\mathbf{x}}, \tag{9}$$

$$\lambda_2 \|\bar{\mathbf{x}}\|_2^2 \leq \mathbf{x}^\top L \mathbf{x} \leq \lambda_n \|\bar{\mathbf{x}}\|_2^2, \tag{10}$$

with λ_n and λ_2 as the largest and smallest non-zero eigenvalue[4] of L, respectively. The results also hold for a weight-balanced (WB) directed graph (digraph) $\mathcal{G}$ by substituting $L_s = \frac{L + L^\top}{2}$ in (9)–(10) with L as the Laplacian of the digraph.

Proof. Following the definition of matrix L, vector $\mathbf{1}_n$ is in the null space of L. Using this, the proof of (9) is straightforward since $\mathbf{1}_n^\top \bar{\mathbf{x}} = 0$. The proof of (10) follows from (9) and the positive definiteness of L. See more details in [56]. □

Corollary 1. *The results of Lemma 5, can be extended for handling two variables $\bar{\mathbf{x}} =: \mathbf{x} - \frac{\mathbf{1}_n^\top \mathbf{x}}{n} \mathbf{1}_n$ and $\bar{\mathbf{y}} =: \mathbf{y} - \frac{\mathbf{1}_n^\top \mathbf{y}}{n} \mathbf{1}_n$ as follows:*

$$\mathbf{x}^\top L \mathbf{y} = \bar{\mathbf{x}}^\top L \bar{\mathbf{y}}, \tag{11}$$

$$\lambda_2 \bar{\mathbf{x}}^\top \bar{\mathbf{y}} \leq \mathbf{x}^\top L \mathbf{y} \leq \lambda_n \bar{\mathbf{x}}^\top \bar{\mathbf{y}}. \tag{12}$$

Note that, for a (edge) union graph $\mathcal{G}_B$, $\lambda_n \leq \lambda_{nB}$ and $\lambda_2 \leq \lambda_{2B}$. Link addition (may) increase the algebraic connectivity of the network [56, 58]. Therefore, given $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2$, we have $\lambda_2(\mathcal{G}) \geq \lambda_2(\mathcal{G}_1)$, $\lambda_2(\mathcal{G}) \geq \lambda_2(\mathcal{G}_2)$. This can be extended, following the Assumption 2, to show $\lambda_2(\mathcal{G}_B(t)) \geq \lambda_2(\mathcal{G}(t))$. One can relate this to the fact that the algebraic connectivity (for $0-1$ adjacency matrices) satisfies $\lambda_2(\mathcal{G}) \geq \frac{1}{n d_g}$ (with d_g as the network diameter) [57, p. 571].

2.3. Time-delay model for the DT agents

In general, multi-agent systems rely on communication networks to exchange information and coordinate actions. These communication networks may introduce delays due to packet processing, transmission time, and network congestion. These delays are typically quantized in integer multiples of the communication time step. Additionally, each node might have its processing time before sending a message or updating its state. These transmission delays between agents can introduce time delays.

Here, we define the time-delay model with k denoting the discrete time index. Following the notation in [59–61], by $\tau_{ij}(k)$ we denote the delay on the transmission link from agent i to agent j at DT step $k = \lfloor \frac{t}{T} \rfloor + 1$ with T as the sampling step and $\lfloor \cdot \rfloor$ as the floor function. If a message sent from agent j at time t_j reaches agent i before time $t_i > t_j$, then $\tau_{ij} = \lceil \frac{t_i}{T} \rceil - \lfloor \frac{t_j}{T} \rfloor - 1$ (with $\lceil \cdot \rceil$ as the ceil function). Therefore, for $t_i - t_j < T$ there is no delay over the link (j, i). However, typically

delays are defined in discrete-time and the sent message at time-step k and received before time-step $k + \tau_{ij} + 1$ implies delay equal to τ_{ij}.

Assumption 3. The followings hold for (integer) time-delay $\tau_{ij}(k)$ on link $(j, i) \in \mathcal{E}(k)$:

(i) $\tau_{ij}(k) \leq \bar{\tau}$, where $1 \leq \bar{\tau} < \infty$ is an integer representing maximum possible delays on the transmission links ($\bar{\tau} = 0$ means no delay). Upper-bound on $\bar{\tau}$ guarantees no lost information and implies that the data from agent i at time k eventually is available to agent j at step $k + \bar{\tau} + 1$ (in finite number of time-steps).

(ii) $\tau_{ij}(k)$ may change at different time-steps k and is heterogeneous for different links, i.e., it may differ for agents and on different links. The time-delays are upper-bounded by some $\bar{\tau}$.

(iii) The transmitted packets over every link are time-stamped and every agent i knows the time t_j at which agent j sent the information over the link (j, i), i.e., the delay $\tau_{ij}(k)$ is known.

(iv) For a shared mutual link between i and j, we consider the same delay for both sides, i.e., $\tau_{ij}(k) = \tau_{ji}(k)$.

(v) At every time k, at least one packet is delivered over the network (possibly with some delay), i.e., at any time step k we have $\tau_{ij}(k) \neq 0$ for at least one pair (i, j) (and subsequently $\tau_{ji}(k) \neq 0$ due to (iv)).

Remark 2. Assumption 3 is less restrictive than many existing literature in distributed sensor networks. To clarify, part (i) only implies no packet loss and data dropout over the network. Part (ii) generalizes the existing literature with fixed delays at all links [56] by considering general heterogeneous delays as in [59,62]. In other words, the delays may differ at different links. Part (iii) is a typical assumption in consensus literature [56] and data transmission networks for example for clock synchronization [63]. Part (iv) implies that both agents i, j process their shared information *simultaneously*.[5] It can be relaxed for asymmetric delays $\tau_{ij} \neq \tau_{ji}$, by considering $\max\{\tau_{ij}, \tau_{ji}\}$ at both sides. This is to fulfil the feasibility condition (as discussed later in Section 6). Part (v) is only to ensure that at every time-step k (at least) two nodes update their states.

Every agent needs to record its previous information at the last $\bar{\tau}$ time-steps to match them with the received information from agent $j \in \mathcal{N}_i$ at the next-coming time-steps. Further, define $\mathcal{I}_{k-r,ij}$ as the *indicator function* capturing the delay $\tau_{ij}(k) \leq \bar{\tau}$ on the link (j, i) as follows,

$$\mathcal{I}_{k,ij}(r) = \begin{cases} 1, & \text{if } \tau_{ij}(k) = r, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Define the *temp graph* $\mathcal{G}^\tau(t) = \{\mathcal{V}, \mathcal{E}^\tau(t)\}$ as the temporary graph representing the neighbourhood of agents at time t (and time-step k) based on the delays τ_{ij}. At time $kT - T < t \leq kT$ if agent i receives a (possibly delayed) packet from j, then $(j, i) \in \mathcal{E}^\tau(t)$ for $kT \leq t < kT + T$, otherwise $(j, i) \notin \mathcal{E}^\tau(t)$.

Remark 3. For switching networks with $B > 0$ in Assumption 2(b), $(\bar{\tau} + 1)T$ needs to be less than switching period of $\mathcal{G}(t)$. This implies that in case of losing a link (i, j) due to a change in $\mathcal{G}(t)$, (at least) one delayed packet from agent i reaches neighbouring agent j before the link (i, j) disappears. Therefore, for any $(i, j) \in \mathcal{E}(t)$ we have $(i, j) \in \mathcal{E}^\tau(t)$, and thus, $\bigcup_{t=k_1 T}^{k_1 T + B} \mathcal{G}^\tau(t) = \bigcup_{t=k_1 T}^{k_1 T + B} \mathcal{G}(t)$. In case of larger $\bar{\tau}$, (if possible) uniform connectivity of $\mathcal{G}_{B^\tau}^\tau(t)$ over a time-window $B^\tau > B$ might be considered in Assumption 2.

---

[4] Recall that in graph theory, for (undirected) connected graphs with link-weight equal to 1, λ_2 is the second smallest eigenvalue of associated Laplacian matrix L and is a real-valued positive number. It is also known as the Fiedler value or Algebraic connectivity [56,57].

[5] The assumption that τ_{ij}, τ_{ji} are known to both agents i, j is well-justified in information-theoretic perspective. This follows from Assumption 3*(iii)* and the assumption that when the packet data leaves the buffer it reaches the receiver with a fixed delay [64]. One may also consider the upper-bound on these delays as a more conservative approach.

3. The proposed continuous-time solution

3.1. Proposed continuous-time dynamics

In this section, we provide two CT 1st-order dynamics to solve problem (3); the solution for problem (2) similarly follows. Following the auxiliary results in the previous section, it is clear that any $\mathbf{x}^*$ for which $\partial f_i^* = \partial f_j^*$ for all i, j must be invariant under the proposed dynamics. To account for nonlinearities, in general, two models can be considered: (i) the *link-based* nonlinearities associated with every link/edge, and (ii) the *node-based* nonlinearities associated with every agent/node. The proposed continuous-time dynamics are based on *local* information available at each agent i and received from its neighbours $\mathcal{N}_i$.

Node-based Nonlinear Solution:

$$\dot{\mathbf{x}}_i = -\sum_{j \in \mathcal{N}_i} W_{ij} g\left(\partial f_i - \partial f_j\right), \qquad (14)$$

Link-based Nonlinear Solution:

$$\dot{\mathbf{x}}_i = -\sum_{j \in \mathcal{N}_i} W_{ij}\left(g(\partial f_i) - g(\partial f_j)\right), \qquad (15)$$

where W_{ij} and $\mathcal{N}_i$ are time-dependent in general (Assumption 2). In the rest of this paper, we discuss the properties of (14)–(15) as the solutions of problem (3). For the sake of notation simplicity, we drop the dependence of W_{ij}, $\mathcal{N}_i$, and ∂f_i on t unless where needed. For $g(x) = x$, (14) and (15) represent the classic linear solution given in [6,46]. However, unlike [46] (and many other papers in the literature) W is not necessarily bi-stochastic but is only symmetric with positive entries.

As compared to many existing linear dynamics proposed in the literature [4,6,14,33,46,47,50,65–68] or ADMM-based solutions proposed in the literature [35–40], this work addresses nonlinearity $g(\cdot)$ at the nodes or the links. This nonlinearity might be due to inherent property of the system, e.g., quantized information exchange among the nodes (see Section 8) or the RRLs in automatic-generation-control setup (see Section 7). In this aspect, the existing simplified linear solutions do not work properly and may result in an optimality gap. On the other hand, our nonlinear model allows for improving the convergence rate and finite/fixed-time convergence by adding sign-based functions as nonlinearity $g(\cdot)$. This cannot be addressed by the existing linear solutions [4,6,14,33,46,47,50,65–68] or dual-based solutions [35–40].

We make the following assumption on the nonlinear mapping $g(\cdot)$.

Assumption 4. Function $g : \mathbb{R} \mapsto \mathbb{R}$ is a nonlinear odd mapping with $\frac{dg}{dx} \neq 0$ at $x = 0$ and $xg(x) > 0$ for $x \neq 0$, i.e., $g(x)$ is strongly sign-preserving. Further, there exists $K_g, \epsilon > 0$ such that $K_g|x| \geq |g(x)| \geq \epsilon|x|$, referred to as the sector-bound conditions.[6]

Many existing nonlinearities satisfy the above assumption. As an example, a monotonically increasing Lipschitz odd function $g(\cdot)$ satisfies the conditions in Assumption 4. Some other examples are discussed in the rest of this section.

Remark 4. Even though Eq. (14) and (15) represent separate nonlinearities at the nodes and the links, the results of this work hold for their combination (with both node and link nonlinearities satisfying Assumption 4). An example is given in Section 8.

3.2. Examples of practical nonlinearities in applications

In this section, we provide some practical nonlinear functions $g(\cdot)$. First, define $y := \partial f_i - \partial f_j$ and the odd function $\mathrm{sgn}^\nu(y)$ as

$$\mathrm{sgn}^\nu(y) := y|y|^{\nu-1}, \qquad (16)$$

where $|\cdot|$ denotes the absolute value and $\nu \geq 0$. For $\nu = 0$, (16) gives the well-known signum function denoted by $\mathrm{sgn}(y)$ for simplicity. The following applications distinguish this work from many existing literature.

Application I: Quantization

$$g_l(y) := \mathrm{sgn}(y) \exp(g_u(\log(|y|))), \qquad (17)$$

with $g_u(y) := \delta\left[\frac{y}{\delta}\right]$ and $\delta > 0$ as the quantization level. In order to allow quantized information processing at agents and transmission links [7], one can substitute logarithmic quantizer $g_l(\cdot)$ into (14) and (15). Note that uniform quantizer $g_u(\cdot)$ is not "strongly" sign-preserving but it is sign-preserving.[7]

Application II: Saturation

$$g_\kappa(y) := \begin{cases} \kappa\,\mathrm{sgn}(y) & |y| > \kappa, \\ y & |y| \leq \kappa, \end{cases} \qquad (18)$$

with $\kappa > 0$ as the saturation (clipping) level. To account for the limited range of sensors/actuators (saturation) and restriction on transfer of analog/digital signals (signal clipping) [24,70], the nonlinear function $g_\kappa(\cdot)$ can be substituted in (14) and (15). This might be due to physical restrictions to follow the limited rate of increase or decrease in the actuation input $\dot{\mathbf{x}}_i$. An example in power grid applications is given in Section 7.

Application III: Finite/Fixed-time Convergence

$$g_f(y) := \mathrm{sgn}^{\nu_1}(y) + \mathrm{sgn}^{\nu_2}(y), \qquad (19)$$

where $0 < \nu_1 < 1$, $1 < \nu_2$ (fixed-time) or $0 < \nu_2 < 1$ (finite-time). Using this type of nonlinear optimization model some existing work on consensus and optimization [9,10,18,25] show that convergence can be achieved in finite/fixed-time. One may even adopt time-varying ν_1, ν_2 as in consensus algorithms [71] to reach convergence over a prescribed time irrespective of the initialization and system parameters.

Application IV: Robustification

$$g_p(y) := \begin{cases} \frac{1-\epsilon}{\epsilon d}\,\mathrm{sgn}(y) & |y| > d \\ 0 & |y| \leq d \end{cases} \qquad (20)$$

with $0 < \epsilon < 1$, $d > 0$. In order to design protocols robust to *high-intensity outliers*, for example in case of communication channels corrupted with impulsive noise [11,28], one can replace the sign-preserving function $g_p(\cdot)$ in (15). A similar model can be considered to suppress impulsive actuation nonlinearities in (14). However, since Eq. (20) is only sign-preserving (not "strongly"), it may result in steady-state bias from the optimizer, i.e., convergence ensures $\mathbf{x}$ to reach a neighbourhood of $\mathbf{x}^*$.

Note that the sign-based nonlinear functions in Applications III-IV are mostly used in CT. The applications of our proposed solutions are not limited to these nonlinear models, but any $g(\cdot)$ satisfying Assumption 4 might be adopted. For example, the composition of mentioned nonlinear mappings (17)–(20) are also a valid choice for $g(\cdot)$ in (14) and (15), or many other sector-bound nonlinearities which satisfy Assumption 4.

---

[6] As we discuss later, the condition $\epsilon|x| \leq |g(x)|$ is needed for exact convergence (sign-preserving versus "strongly" sign-preserving) and $|g(x)| \leq K_g|x|$ is only needed for the DT case.

[7] Such uniformly quantized dynamics may result in (steady-state) residual and bias from the equilibrium, where the bias (residual) scales with the quantization level δ and gets arbitrarily close to zero for sufficiently small δ. See [20,21,43,69] for some discussions on quantized discrete-time consensus.

4. Convergence analysis in continuous-time

In this section, we analyse the convergence of the CT dynamics (14) and (15) to the optimal value of constrained optimization (3). We first check the feasibility and uniqueness of the solutions under the given dynamics and then prove the convergence.

Lemma 6 (*Feasibility*). *Suppose that Assumptions 2 and 4 hold. Initializing by any* $\mathbf{x}_0 \in S_b$, *the state of agents remain feasible under the CT dynamics* (14) *and* (15)*, i.e.,* $\mathbf{x}(t) \in S_b$ *for all* $t > 0$.

Proof. The proof of feasibility for CT dynamics (14) is given in [49]. For (15) similarly we have,

$$\frac{d}{dt}(\mathbf{x}^\top \mathbf{1}_n) = -\sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} W_{ij}\Big(g(\partial f_i) - g(\partial f_j)\Big). \tag{21}$$

From Assumptions 2 and 4 we have,

$$W_{ij}\Big(g(\partial f_i) - g(\partial f_j)\Big) = -W_{ji}\Big(g(\partial f_i) - g(\partial f_j)\Big).$$

which implies that $\frac{d}{dt}(\mathbf{x}^\top \mathbf{1}_n) = 0$ in (21) and $\mathbf{x}^\top \mathbf{1}_n$ remains time-invariant. Therefore, any feasible initialization $\mathbf{x}_0^\top \mathbf{1}_n = b$ gives feasibility over time $\mathbf{x}(t)^\top \mathbf{1}_n = b$ and $\mathbf{x}(t) \in S_b$ for all $t > 0$. □

In contrast to primal–dual solutions such as ADMM-based methods [35–40], Lemma 6 proves all-time feasibility of our proposed solution. This means that at any termination time of the algorithm the solution preserves feasibility while in the existing ADMM solutions [34–40] there might be feasibility gap that converges to zero asymptotically. This implies that the solution by [35–40] must be fast enough to gain feasibility before the termination of the algorithm.

Note that for any $\mathbf{x}$ satisfying $\nabla F(\mathbf{x}) = \varphi \mathbf{1}_n$, we have $\dot{x}_i = 0$ at every agent i. Therefore, such $\mathbf{x}$ is an equilibrium (invariant-state) of the solution dynamics (14) and (15). In the next theorem, we show that such $\mathbf{x}^*$ is unique for both proposed dynamics. Note that we assume $\mathbf{x}^*$ satisfies the local box constraints (if there are any).

Theorem 1 (*Uniqueness in CT*). *Suppose that Assumptions 1, 2 and 4 hold. Let* $\mathbf{x}^*$ *denote the equilibrium under CT dynamics* (14) *and* (15)*. Then,* $\nabla F(\mathbf{x}^*) = \varphi^* \mathbf{1}_n$ *with* $\varphi^* \in \mathbb{R}$.

Proof. By contradiction assume $\nabla F(\mathbf{x}^*) = (\Lambda_1^*; \dots; \Lambda_n^*)$ where $\Lambda^* \neq \varphi^*$ and for (at least) two agents i, j,

$$\Lambda_i^* \neq \Lambda_j^* \iff \partial f_i^* \neq \partial f_j^*, \tag{22}$$

Define two agents α, β as,

$$\alpha = \underset{q \in \{1, \dots, n\}}{\arg\max} \Lambda_q^*, \quad \beta = \underset{q \in \{1, \dots, n\}}{\arg\min} \Lambda_q^*. \tag{23}$$

From Assumption 2, the union graph $\mathcal{G}_B(t)$ is connected for every $t \geq 0$, and therefore, there is a path in $\mathcal{G}_B(t)$ from agent (node) α to every other agent (node) including β. In this path, one can find (at least) two nodes $\overline{\alpha}$ and $\overline{\beta}$ with the set of neighbours $\mathcal{N}_{\overline{\alpha}}$ and $\mathcal{N}_{\overline{\beta}}$, respectively, such that

$$\Lambda_{\overline{\alpha}}^* \geq \Lambda_{\mathcal{N}_{\overline{\alpha}}}^*, \quad \Lambda_{\overline{\beta}}^* \leq \Lambda_{\mathcal{N}_{\overline{\beta}}}^*, \tag{24}$$

where the strict inequalities in the above hold for (at least) one node in $\mathcal{N}_{\overline{\alpha}}$ and one node in $\mathcal{N}_{\overline{\beta}}$. Therefore, from Assumption 4, $\dot{x}_{\overline{\alpha}}^* < 0$ and $\dot{x}_{\overline{\beta}}^* > 0$ over any time-window of length B (for both proposed dynamics). This contradicts our equilibrium assumption $\dot{\mathbf{x}}^* = \mathbf{0}$. □

In the virtue of Lemmas 4, 6, and Theorem 1, for a feasible initial state $\mathbf{x}_0 \in S_b$ there exist only one equilibrium $\mathbf{x}^*$ satisfying $\nabla F(\mathbf{x}^*) = \varphi^* \mathbf{1}_n$ under dynamics (14) and (15). In order to prove convergence to $\mathbf{x}^*$, the following lemma is needed.

Lemma 7 ([10]). *For* $g(\cdot)$ *and* W *satisfying Assumption 2, 4 and* $\psi_i, \psi_j \in \mathbb{R}$,

$$\sum_{i=1}^{n} \psi_i \sum_{j=1}^{n} W_{ij} g(\psi_j - \psi_i) = \sum_{i,j=1}^{n} \frac{W_{ij}}{2}(\psi_j - \psi_i)g(\psi_j - \psi_i),$$

Similarly,

$$\sum_{i=1}^{n} \psi_i \sum_{j=1}^{n} W_{ij}(g(\psi_j) - g(\psi_i))$$

$$= \sum_{i,j=1}^{n} \frac{W_{ij}}{2}(\psi_j - \psi_i)(g(\psi_j) - g(\psi_i)).$$

Theorem 2 (*Convergence*). *Suppose that Assumption 1, 2, 4 hold and* $\mathbf{x}_0 \in S_b$. *The proposed CT dynamics* (14) *and* (15) *converge to the optimal value of* (3) *denoted by* $\mathbf{x}^*$ *for which* $\nabla F(\mathbf{x}^*) \in \text{span}\{\mathbf{1}_n\}$, *or simply,*

$$\exists \varphi^* \in \mathbb{R}, \quad \partial f_i^* = \partial f_j^* = \varphi^*.$$

Proof. The proof for dynamics (14) is given in the proof of [49, Theorem 2]. Similar proofs hold for dynamics (15) with Lyapunov function $\overline{F}(\mathbf{x}) := F(\mathbf{x}) - F(\mathbf{x}^*)$ and recalling from Lemma 7 that $\Big(\partial f_i - \partial f_j\Big)\Big(g(\partial f_i) - g(\partial f_j)\Big) > 0$. □

It is worth mentioning that the existing literature [4,6,14,46,47,50,65–68] mostly work over all-time connected networks, while, from Theorem 1 and 2, our proposed solution works over uniformly-connected networks that may lose connectivity over some time-intervals. Note that from Theorem 1 to prove unique equilibrium of the solution we only need the union network $\mathcal{G}_B(t)$ to be connected over some interval B. This makes our solution applicable over mobile multi-agent systems where the network is dynamic and the connectivity might be lost temporarily. This is in contrast to many existing solutions [4,6,14,46, 47,50,65–68] where the network is static and/or all-time connected.

5. The proposed solution in discrete-time

In this section, first, we provide the discrete-time version of (14)–(15) respectively as,

$$x_i(k + 1) = x_i(k) - T \sum_{j \in \mathcal{N}_i} W_{ij} g\Big(\partial f_i(k) - \partial f_j(k)\Big), \tag{25}$$

$$x_i(k + 1) = x_i(k)$$
$$\quad - T \sum_{j \in \mathcal{N}_i} W_{ij}\Big(g(\partial f_i(k)) - g\Big(\partial f_j(k)\Big)\Big), \tag{26}$$

with $k \geq 1$ and T as the time-step.

In the rest of the paper, we use a simplifying notation. Define the *anti-symmetric* matrices $\mathcal{D}(k) = [D_{ij}(k)]$ and $\mathcal{D}^g(k) = [D_{ij}^g(k)]$, as the weighted difference of the gradients over all links, i.e.,

$$D_{ij}(k) = \partial f_i(k) - \partial f_j(k), \tag{27}$$

$$D_{ij}^g(k) = g(\partial f_i(k)) - g(\partial f_j(k)). \tag{28}$$

The following theorem gives the proof of convergence assuming sufficiently small T (as defined later in Section 5.1).

Theorem 3 (*DT Feasibility/Uniqueness/Convergence*). *Under Assumption 1, 2, 4 and initializing from a feasible point* $\mathbf{x}_0 \in S_b$, *protocols* (25) *and* (26) *converge to the feasible unique equilibrium point* $\mathbf{x}^*$ *in the form* $\nabla F(\mathbf{x}^*) = \varphi^* \mathbf{1}_n$ *(with* $\varphi^* \in \mathbb{R}$) *for sufficiently small (sampling) step* $T < T_\lambda$ *(* T_λ *is defined later in* (44)*)*.

Proof. Following (25) and (26), we get

$$\sum_{i=1}^{n} x_i(k+1) = \sum_{i=1}^{n} x_i(k) - \sum_{i=1}^{n} T \sum_{j \in \mathcal{N}_i} W_{ij} g(D_{ij}(k)). \tag{29}$$

$$\sum_{i=1}^{n} x_i(k+1) = \sum_{i=1}^{n} x_i(k) - \sum_{i=1}^{n} T \sum_{j \in \mathcal{N}_i} W_{ij} \mathcal{D}_{ij}^g(k). \tag{30}$$

For any link (i,j) and (j,i) in $\mathcal{G}(k)$, under the given assumptions we have $W_{ij} = W_{ji}$, $\mathcal{D}_{ij}^g(k) = -\mathcal{D}_{ji}^g(k)$, and $g(\mathcal{D}_{ij}(k)) = -g(\mathcal{D}_{ji}(k))$, which implies that,

$$\sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} W_{ij} g(\mathcal{D}_{ij}(k)) = 0, \tag{31}$$

$$\sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} W_{ij} \mathcal{D}_{ij}^g(k) = 0, \tag{32}$$

and therefore, $\sum_{i=1}^{n} x_i(k+1) = \sum_{i=1}^{n} x_i(k)$ for all $k > 0$. Therefore, initializing from $\mathbf{x}_0 \in S_b$ we have $\mathbf{x}(k) \in S_b$, and the feasibility for both DT protocols (48) and (49) follows. It should be mentioned that the feasibility condition for the link-based nonlinear solution holds for general weight-balanced directed networks. The proof of uniqueness follows similar procedure as in Theorem 1, where instead of having $\dot{\mathbf{x}}_{\alpha}^* < 0$ and $\dot{\mathbf{x}}_{\beta}^* > 0$ in the CT case, we have $\mathbf{x}_{\alpha}^*(k+1) - \mathbf{x}_{\alpha}^*(k) < 0$ and $\mathbf{x}_{\beta}^*(k+1) - \mathbf{x}_{\beta}^*(k) > 0$ over any time-window of length B. The proof of convergence and the bound on T is discussed later in Section 5.1. $\square$

Remark 5. For the link-based protocol (26), one can extend the solution to weight-balanced (WB) directed networks. This is because in the proof of Theorem 3, one can restate Eq. (32) for $\sum_{i=1}^{n} W_{ij} = \sum_{j=1}^{n} W_{ji}$ over WB digraphs that proves feasibility and convergence as Lemma 7 also holds for WB digraphs. Note that proof of uniqueness is irrespective of the weights and only depends on the network connectivity. Similar reasoning proves feasibility in Lemma 6 and convergence for CT dynamics (15) over WB digraphs. Distributed (weight) balancing algorithms can be adopted to design such directed networks [72].

5.1. Rate of convergence

Next, to determine the rate of convergence of the proposed (delay-free) protocols, we further make the following assumption.

Assumption 5. For the local cost functions $f_i(x_i) : \mathbb{R} \mapsto \mathbb{R}$,

- there exists $u < \infty$ such that $\partial^2 f_i(x_i) < 2u$. This implies that $\partial f_i(x_i)$ are Lipschitz continuous.
- (Strong-convexity) there exists $v > 0$ such that $\partial^2 f_i(x_i) > 2v$.

To incorporate a non-smooth penalty term or barrier function f_i^σ for the box constraints [29,49], e.g., $[u]^+ = \max\{u,0\}^\sigma$ for $\sigma = 1$, one can replace it with smooth equivalent $\mathcal{L}(u,\mu) = \frac{1}{\mu} \log(1 + \exp(\mu u))$. This is a typical reformulation in machine learning literature [1,73]. It is known that $\mathcal{L}(z,\mu) - \max\{z,0\} \le \frac{1}{\mu}$ and the two functions can become arbitrarily close by sufficiently large μ [73]. $[u]^+$ with $\sigma \ge 2$ is another alternative smooth option [74]. Assumption 5 helps to explicitly derive a *sufficient* bound on the sampling step to ensure convergence. Based on this assumption, we reformulate Lemma 2 as follows.

Lemma 8. Assume the function $f_i : \mathbb{R} \mapsto \mathbb{R}$ to be strongly convex and $2v < \partial^2 f_i(x_i) < 2u$ (Assumption 5). Then, for two feasible points $\mathbf{x}(k+1), \mathbf{x}(k) \in S_b$, and $\Delta \mathbf{x} =: \mathbf{x}(k+1) - \mathbf{x}(k)$,

$$F(\mathbf{x}(k+1)) > F(\mathbf{x}(k)) + \overline{\nabla F}_1^\top \Delta \mathbf{x} + v \Delta \mathbf{x}^\top \Delta \mathbf{x}, \tag{33}$$

$$F(\mathbf{x}(k+1)) > F(\mathbf{x}(k)) - \frac{1}{4v} \overline{\nabla F}_1^\top \overline{\nabla F}_1, \tag{34}$$

and similarly,

$$F(\mathbf{x}(k+1)) < F(\mathbf{x}(k)) + \overline{\nabla F}_1^\top \Delta \mathbf{x} + u \Delta \mathbf{x}^\top \Delta \mathbf{x}, \tag{35}$$

$$F(\mathbf{x}(k+1)) < F(\mathbf{x}(k)) - \frac{1}{4u} \overline{\nabla F}_1^\top \overline{\nabla F}_1, \tag{36}$$

with gradient dispersion defined as $\overline{\nabla F}_1 := \nabla F(\mathbf{x}(k+1)) - \frac{1}{n} (\mathbf{1}_n^\top \nabla F(\mathbf{x}(k+1))) \mathbf{1}_n$.

Proof. The proof follows from Lemma 2. See details in [6]. $\square$

Corollary 2. As a direct result of Lemma 8, substituting $\mathbf{x}_2 = \mathbf{x}^*$ in (34) and (36), for any feasible $\mathbf{x} \in S_b$ and $\overline{F}(k) := F(k) - F^*$,

$$\frac{1}{4u} \overline{\nabla F}^\top \overline{\nabla F} \le \overline{F}(\mathbf{x}) \le \frac{1}{4v} \overline{\nabla F}^\top \overline{\nabla F}. \tag{37}$$

The above corollary follows from the strong convexity of $\overline{F}$ and holds for general allocation dynamics; for example, a similar statement is given for the linear solution in [6]. Using Assumption 5 and Lemma 8, one can find the limit on the convergence rate under the proposed CT and DT protocols (assuming no latency, i.e., $\overline{\tau} = 0$). Recall that, for the CT case, $\dot{\overline{F}} = \nabla F^\top \dot{\mathbf{x}}$. In the following, we state the results for (14) and the solution for (15) similarly follows. Using the definition of Laplacian in (1), one can rewrite (14) (with some abuse of notation) as $\dot{\mathbf{x}} = -L g(\nabla F)$. Substituting this for $\dot{\mathbf{x}}$ and from Assumption 4 and Corollary 1,

$$-K_g \nabla F^\top L \nabla F \le \dot{\overline{F}}(\mathbf{x}) \le -\varepsilon \nabla F^\top L \nabla F, \tag{38}$$

where we used the following inequalities from Assumption 4,

$$\varepsilon |(\partial f_i(k)| < |g(\partial f_i(k))| < K_g |(\partial f_i(k))|, \tag{39}$$

$$\varepsilon |\mathcal{D}_{ij}(k)| < |g(\mathcal{D}_{ij}(k))| < K_g |\mathcal{D}_{ij}(k)|. \tag{40}$$

For now, assume $B = 0$ and later we extend it to $B > 0$. Using (10) with λ_2 described in Lemma 5 and recalling the notation $\overline{\nabla F}$ in Lemma 8, we have

$$-K_g \lambda_n \overline{\nabla F}^\top \overline{\nabla F} \le \dot{\overline{F}} \le -\varepsilon \lambda_2 \overline{\nabla F}^\top \overline{\nabla F}, \tag{41}$$

and from (37)

$$-4u K_g \lambda_n \overline{F}(\mathbf{x}) \le \dot{\overline{F}} \le -4v \varepsilon \lambda_2 \overline{F}(\mathbf{x}), \tag{42}$$

which is consistent with the linear case ($\varepsilon = K_g = 1$) in [6]. Recall that for $\overline{F} = 0$ we also have $\dot{\overline{F}} = 0$. Next, for the DT case, following Eq. (35),

$$\overline{F}(k+1) \le \overline{F}(k) + \nabla F(k)^\top \Delta \mathbf{x} + u \Delta \mathbf{x}^\top \Delta \mathbf{x},$$

with $\Delta \mathbf{x} := \mathbf{x}(k+1) - \mathbf{x}(k)$. To satisfy $\overline{F}(k+1) \le \overline{F}(k)$, we need,

$$\nabla F(k)^\top \Delta \mathbf{x} + u \Delta \mathbf{x}^\top \Delta \mathbf{x} \le 0.$$

From Assumption 4 and following a similar line of reasoning to get Eq. (42), the above is satisfied if,

$$-\varepsilon T \lambda_2 \overline{\nabla F}^\top \overline{\nabla F} + u K_g^2 T^2 \overline{\nabla F}^\top L^\top L \overline{\nabla F} \le 0,$$

where we substituted $\Delta \mathbf{x} = L \nabla F = L \overline{\nabla F}$ and used Lemma 5 and Corollary 1. From the same Lemma 5 we have $\overline{\nabla F}^\top L^\top L \overline{\nabla F} \le \lambda_n^2 \overline{\nabla F}^\top \overline{\nabla F}$. Therefore, the sufficient condition for convergence is,

$$(u K_g^2 T \lambda_n^2 - \varepsilon \lambda_2) \overline{\nabla F}^\top \overline{\nabla F} \le 0, \tag{43}$$

which gives the sufficient bound on T as,

$$T \le \frac{\varepsilon \lambda_2}{u K_g^2 \lambda_n^2} =: T_\lambda. \tag{44}$$

Then, the upper bound on the convergence rate of the residual follows from Lemma 8 and (37) as,

$$\frac{\overline{F}(k+1)}{\overline{F}(k)} \le 1 + 4v(u K_g^2 T^2 \lambda_n^2 - T \lambda_2 \varepsilon). \tag{45}$$

The above gives an estimate on the (linear) convergence rate of $\overline{F}$ for sufficiently small T satisfying (44) and also proves the convergence in Theorem 3. For the quadratic cost functions (as in the economic dispatch problem), we have $u = v$, and the above equations can be more simplified. Further, in case of B-connectivity instead of all-time connectivity ($B = 1$), one can derive Eq. (45) for $\frac{\overline{F}(k+B)}{\overline{F}(k)}$ over union graph $\mathcal{G}_B$ and modify (44) as $T B \le T_\lambda$ with λ_2, λ_n as the eigenvalues of $\mathcal{G}_B$.
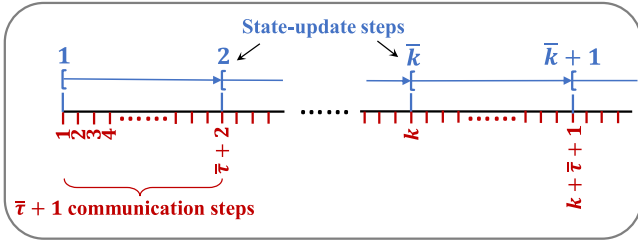
Fig. 1. This figure illustrates Case I to handle delays over the network. The original communication time-scale k (red) versus the state-update time-scale $\overline{k}$ (blue) are shown. The information is shared over the longer time scale $\overline{k}$ to update the state $x_i(\overline{k})$ via the dynamics (46) and (47). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Remark 6. Recall that, from Lemma 5 for WB directed graphs, values λ_2, λ_n in (43) (and in the subsequent equations) denote the eigenvalues of the symmetric matrix L_s instead of L.

6. Networks with time-delays

Next, we extend the solutions to the time-delayed case. We consider two approaches to overcome network latency according to the following remark.

Remark 7. Following Assumption 3, the knowledge of every τ_{ij} is key to satisfy the feasibility condition in the delayed case, see (31) and (32). Note that $\mathcal{D}(k-r)$ and $\mathcal{D}^g(k-r)$ (with r as the delay) needs to be anti-symmetric. This follows from Assumption 3 and Remark 2. In other words, every agent i knows the delay r to match the received $g(\partial f_j(k-r))$ with its own previous information $g(\partial f_i(k-r))$ to find $\mathcal{D}^g_{ij}(k-r)$, and the same holds for agent j to find $\mathcal{D}^g_{ji}(k-r)$.

6.1. Case I : information-update over a longer time-scale

A straightforward solution in the delayed case is to have the agents wait for $\overline{\tau}$ steps (with $\overline{\tau}$ as the maximum possible delay) until they collect (at least) one delayed packet from the neighbouring agents (before processing for the next iteration) and then update their state.[8] This alternative approach requires knowledge of $\overline{\tau}$ (or an upper bound). Obviously, due to the agents' slower update rate, this approach's convergence rate is low. In this case, we define a new time-scale $\overline{k} = \lfloor \frac{k-1}{\overline{\tau}+1} \rfloor + 1$ (see Fig. 1) and update the state of each agent i as follows,

$$x_i(\overline{k}+1) = x_i(\overline{k}) - T \sum_{j \in \mathcal{N}_i} W_{ij} g(\mathcal{D}_{ij}(\overline{k})), \tag{46}$$

$$x_i(\overline{k}+1) = x_i(\overline{k}) - T \sum_{j \in \mathcal{N}_i} W_{ij} \mathcal{D}^g_{ij}(\overline{k}). \tag{47}$$

Over this $\overline{\tau}+1$ time-steps (i.e., two consecutive update steps $\overline{k}_1$ and $\overline{k}_2$) on every link (i,j), every agent sends 1 message at step $\overline{k}$ and receives the messages from $j \in \mathcal{N}_i$ by step $\overline{k}+1$,[9] see Algorithm 1. This is used in the proof of convergence.

Theorem 4 (Feasibility/Uniqueness/Convergence). *Under Assumptions 1–4 and initializing from a feasible point $\mathbf{x}_0 \in S_b$, protocols (46) and (47) converge to the feasible unique equilibrium point $\mathbf{x}^*$ in the form $\nabla F(\mathbf{x}^*) = \varphi^* \otimes \mathbf{1}_n$ for sufficiently small T satisfying $T < T_\lambda$ (with T_λ given as (44)).*

---
[8] Similar solution to handle delays in consensus protocols is discussed in [59, Remark 3].

[9] In an equivalent setup, agents may send their messages per scale k where, at the update scale $\overline{k}$, receive at least 1 and at most $\overline{\tau}$ messages. Then, the step size T needs to be down-scaled accordingly to satisfy the convergence criteria.

Algorithm 1 Delay-Tolerant solution: longer time-scale
1: **Input:** W_{ij}s, $\mathcal{N}_i$, $f_i(\cdot)$, T, $\overline{\tau}$
2: **Initialization:** Set $\overline{k} = k = 1$, randomly allocate feasible $\mathbf{x}(1)$ (e.g., via [6, Algorithm 2])
3: Node i shares $\partial f_i(1)$ with neighbours $j \in \mathcal{N}_i$
4: **While** termination criteria NOT true **do**
5: Set $k \leftarrow k + 1$
6: **If** $\frac{k-1}{\overline{\tau}+1} = \overline{k}$
7: Node i calculates $\mathcal{D}^g_{ij}(\overline{k})$ (or $\mathcal{D}_{ij}(\overline{k})$) based on received data $g(\partial f_j(\overline{k}))$ (or $\partial f_j(\overline{k})$) from $j \in \mathcal{N}_i$
8: Updates $x_i(\overline{k}+1)$ via Eq. (47) (or (46))
9: Shares $\partial f_i(\overline{k}+1)$ with neighbours $j \in \mathcal{N}_i$
10: **End If**
11: Set $\overline{k} \leftarrow \lfloor \frac{k-1}{\overline{\tau}+1} \rfloor + 1$
12: **End While**
13: **Output:** x_i^*, $F(\mathbf{x}^*) = \sum_{i=1}^n f_i(x_i^*)$

Proof. The proof of feasibility and uniqueness follows a similar procedure as in the proof of Theorem 3 considering the longer time-scale $\overline{k}$. Since at every update step $\overline{k}$ only 1 message is received from every agent in $\mathcal{N}_i$, following similar expressions to derive Eq. (44), the same criteria T_λ ensures convergence. □

6.2. Case II : using all delayed data at the same time-scale

In the most general time-delayed version of the DT protocols (25) and (26), agent i updates its state based on all available (received) information as,

$$x_i(k+1) = x_i(k)$$
$$- T \sum_{j \in \mathcal{N}_i} \sum_{r=0}^{\overline{\tau}} W_{ij} g(\mathcal{D}_{ij}(k-r)) \mathcal{I}_{k-r,ij}(r), \tag{48}$$

$$x_i(k+1) = x_i(k).$$
$$- T \sum_{j \in \mathcal{N}_i} \sum_{r=0}^{\overline{\tau}} W_{ij} \mathcal{D}^g_{ij}(k-r) \mathcal{I}_{k-r,ij}(r), \tag{49}$$

Note that in (48), $\mathcal{D}_{ij}(k-r)$ can be easily calculated as agent i records all its gradients $\partial f_i(k)$ at the last $\overline{\tau}$ steps and knows the time delay r of the received *time-stamped* message $\partial f_j(k-r)$. In fact, the second $\sum$ sums all the gradient differences based on all received data from $\mathcal{N}_i$ at step k, i.e., for all pairs of $\mathcal{D}_{ij}(s)$ and s satisfying,[10]

$$\{k - \overline{\tau} \leq s \leq k, s + \tau_{ij}(s) = k, j \in \mathcal{N}_i\}. \tag{50}$$

or equivalently,

$$\{\mathcal{I}_{k-s,ij} = s, j \in \mathcal{N}_i\}. \tag{51}$$

Similar arguments hold for $\mathcal{D}^g_{ij}(k-r)$. Further, for Lipschitz $g(\cdot)$ with constant $K_g < \infty$ (and smooth $f_i(\cdot)$), from Definition 1, we have $\mathcal{D}^g_{ij}(k-r) \leq K_g \mathcal{D}_{ij}(k-r)$.

Remark 8. Following Remark 3, for switching network $\mathcal{G}(k)$, the switching period needs to be longer than $(\overline{\tau}+1)T$ (or $\overline{\tau}+1$ time-steps). This assumption implies that agent i receives at least one (possibly) delayed packet from every neighbour in $\mathcal{N}_i$ before losing its link due to network variation (ensuring Assumption 3(v)). Therefore, following the B-connectivity in Assumption 2, we need $(\overline{\tau}+1)T < B$.

---
[10] Recall that the processed information at the time-step k are received in the time-period $(kT - T, T]$. Therefore, (51) can be rewritten as $\{k - \overline{\tau} \leq s \leq k, kT - T < sT + \tau_{ij}(s)T \leq kT, j \in \mathcal{N}_i\}$ in the asynchronous case, for example, the delays could be positive real values (instead of integers).

Algorithm 2 Delay-Tolerant solution: the same time-scale
___
1: **Input:** W_{ij}s, $\mathcal{N}_i$, $f_i(\cdot)$, T, $\overline{\tau}$
2: **Initialization:** Set $k = 1$, randomly allocate feasible $\mathbf{x}(1)$ (e.g., via [6, Algorithm 2])
3: **While** termination criteria NOT true **do**
4:     Node i receives time-stamped $g(\partial f_j(k-\overline{\tau}_{ij}))$ (resp. $\partial f_j(k-\overline{\tau}_{ij})))$ from $j \in \mathcal{N}_i$
5:     Records $\mathcal{D}^g_{ij}(k-\overline{\tau}_{ij})$ (resp. $\mathcal{D}_{ij}(k-\overline{\tau}_{ij})$) based on known delay $\overline{\tau}_{ij}$
6:     Updates $x_i(k+1)$ via Eq. (49) (resp. (48))
7:     Shares $\partial f_i(k+1)$ with neighbours $j \in \mathcal{N}_i$
8:     Sets $k \leftarrow k + 1$
9: **End While**
10: **Output:** x_i^*, $F(\mathbf{x}^*) = \sum_{i=1}^n f_i(x_i^*)$
___

Note that for the update at every step k, Assumption 3(*v*) says that at least one package over the network is delivered. Otherwise, $x_i(k) = x_i(k-1)$ for all i and no update occurs. The solution is summarized in Algorithm 2. The feasibility/uniqueness under (48) and (49) follows similar to the delay-free case.

Lemma 9 (*Feasibility/Uniqueness*). *Let Assumptions 1–4 hold. Initializing from feasible states $\mathbf{x}_0 \in S_b$, under DT protocols (48) and (49), $\mathbf{x}(k) \in S_b$. Further, the unique equilibrium point $\mathbf{x}^*$ satisfies $\nabla F(\mathbf{x}^*) = \varphi^* \mathbf{1}_n$.*

Proof. Following (48) and (49), we get

$$\sum_{i=1}^n x_i(k+1) = \sum_{i=1}^n x_i(k)$$
$$- \sum_{i=1}^n T \sum_{j \in \mathcal{N}_i} \sum_{r=0}^{\overline{\tau}} W_{ij} g(\mathcal{D}_{ij}(k-r)) \mathcal{I}_{k-r,ij}(r). \qquad (52)$$

$$\sum_{i=1}^n x_i(k+1) = \sum_{i=1}^n x_i(k)$$
$$- \sum_{i=1}^n T \sum_{j \in \mathcal{N}_i} \sum_{r=0}^{\overline{\tau}} W_{ij} \mathcal{D}^g_{ij}(k-r) \mathcal{I}_{k-r,ij}(r). \qquad (53)$$

For any link (i,j) and (j,i) in $\mathcal{G}(k)$, from Assumption 2, 3, and 4 we have $W_{ij} = W_{ji}$, $\mathcal{D}^g_{ij}(k-r) = -\mathcal{D}^g_{ji}(k-r)$, $g(\mathcal{D}_{ij}(k-r)) = -g(\mathcal{D}_{ji}(k-r))$, and $\mathcal{I}_{k-r,ij}(r) = \mathcal{I}_{k-r,ji}(r)$ for $0 \le r \le \overline{\tau}$. This implies that,

$$\sum_{i=1}^n T \sum_{j \in \mathcal{N}_i} \sum_{r=0}^{\overline{\tau}} W_{ij} g(\mathcal{D}_{ij}(k-r)) \mathcal{I}_{k-r,ij}(r) = 0,$$

$$\sum_{i=1}^n T \sum_{j \in \mathcal{N}_i} \sum_{r=0}^{\overline{\tau}} W_{ij} \mathcal{D}^g_{ij}(k-r) \mathcal{I}_{k-r,ij}(r) = 0,$$

and therefore, $\sum_{i=1}^n x_i(k+1) = \sum_{i=1}^n x_i(k)$ for all $k \ge 1$. Therefore, initializing from $\mathbf{x}_0 \in S_b$ we have $\mathbf{x}(k) \in S_b$ under both (48) and (49). This proves the feasibility. The proof of uniqueness follows similar reasoning as in the proof of Theorem 3 and 4 over the uniformly connected network $\mathcal{G}_B(t)$ (or $\mathcal{G}^\tau_{B\tau}(t)$ in Remark 3). $\square$

Remark 9. Case I is more practical than Case II in applications with low capacity/buffer at the computing nodes. This is because at time $k = \overline{k}$ node i sends (and receives) one message to (and from) the nodes $\mathcal{N}_i$. Note that, for large delays (and the same T), this solution converges (although with a low rate) while Case II may not necessarily converge. Further, for unknown (or large) $\overline{\tau}$, Case II requires a high-capacity memory/buffer at the nodes to record the previous information (on the gradients). In case of limited (m-slot) memory/buffer, the nodes may record and use a portion (last m) of the previous states instead, and discard the rest before $k - m$ (since they are time-stamped). This simply

implies losing some links over $\mathcal{G}^\tau_{B}$,[11] and follows from Remark 1 and 3 and Assumption 2.

Theorem 5 (*Sufficiency*). *Under Assumptions 1–4, and initializing from feasible state $\mathbf{x}_0 \in S_b$, the proposed protocols (48)–(49) converge to the optimal solution of (3) for $T(\overline{\tau} + 1) < T_\lambda$.*

Proof. First, consider homogeneous delays $\tau_{ij} = \overline{\tau}$, where agents' states at any time-step k get updated next at $k + \overline{\tau} + 1$ and every $\overline{\tau} + 1$ steps afterwards (see Fig. 1). For this case, following Theorem 3, $T < T_\lambda$ ensures the convergence. For general heterogeneous time-varying delays, $\Delta \mathbf{x}$ needs to be scaled by $\overline{\tau} + 1$, and accordingly, T_λ is down-scaled by $\overline{\tau} + 1$ to guarantee convergence. Consider two cases: (i) for time-invariant delays $\sum_{r=0}^{\overline{\tau}} \mathcal{I}_{k-r,ij}(r) = 1$ in (48), implying that *only one* delayed packet is received from every $j \in \mathcal{N}_i$. This implies the same bound as $T < T_\lambda$. For time-varying delays, from (13), $0 < \sum_{r=0}^{\overline{\tau}} \mathcal{I}_{k-r,ij}(r) \le (\overline{\tau} + 1)$. This implies that $\Delta \mathbf{x}$ is scaled by $\overline{\tau} + 1$ and accordingly, from (29), T needs to be down-scaled by $\overline{\tau} + 1$, i.e., $T(\overline{\tau} + 1) < T_\lambda$. The proof for dynamics (49) similarly follows. $\square$

For time-varying but equi-probable delays $r = 0, 1, \ldots, \overline{\tau}$ in (13), one can claim that (in average) over $\overline{\tau} + 1$ trials node i receives one message from $j \in \mathcal{N}_i$ (at every k) and the bound is $T < T_\lambda$.

Communication time-delay over the network is not addressed by the existing solutions [4,6,14,33,46,47,50,65–68]. However, latency is a common issue in many multi-agent systems including the distributed resource allocation setup. Therefore, most existing works in the literature may not work properly in the presence of time-delays in the data-transmission network. The mentioned literature may lose resource-demand feasibility and/or result in some optimality gap in the presence of time-delays. In this aspect, our proposed delay-tolerant model advances the state-of-the-art and provides solutions that can be implemented in practice.

7. Simulation in distributed scheduling setup

In this section, we consider resource allocation over the power generation networks and the smart grid, known as the *economic dispatch problem* (EDP) [6,26,75]. The objective is to allocate optimal power outputs to the electricity generators to supply the *load demand D* (in MW) and to minimize the operating costs. In [4,26,41,47,76] this cost function is given as,

$$F(\mathbf{x}) = \sum_{i=1}^n \gamma_i x_i^2 + \beta_i x_i + \alpha_i, \quad \sum_{i=1}^n x_i = D, m_i \le x_i \le M_i, \qquad (54)$$

where $x_i \in \mathbb{R}$ is the output power at generator i. To include the box constraints, penalty terms are considered [49]. The parameters in the cost function (54) are defined based on the type of the power generators (coal-fired, oil-fired, nuclear, etc.), for example, see Table 2. The other parameter values are: $\alpha_i = 0$ and $m_i = 20$. The min and max RRL values given in [77] are equal to 1 and 3 MW/min (for oil/coal-type generators).

In a more complicated setup, such a quadratic cost model is defined to adjust the power-demand mismatch over the grid (e.g., due to generator outage), known as the *Automatic Generation Control (AGC)* problem [49]. Given a known power mismatch, the idea is to allocate enough power to the generators to compensate for it while minimizing the power deviation cost. In this setup, the generators are subject to an extra physical constraint, known as ramp-rate-limit (RRL). This implies that the rate of increase or decrease of the produced power is constrained within certain limits and the generators cannot freely speed up/down their power generation. Such nonlinear constraints are

___
[11] Convergence over lossy networks, following from Assumption 2 and Remark 1 is another promising direction of our future research.

Table 2
Parameters of the generator cost (in $) and its maximum power (in MW) for different types [4,41,76].

Type	M_i	β_i	γ_i
A	80	2.0	0.04
B	90	3.0	0.03
C	70	4.0	0.035
D	70	4.0	0.03
E	80	2.5	0.04

Table 3
The elapsed time (in ms) to run one iteration (in average) for each scenario in Fig. 2. This table gives a measure to compare the computational complexity of the algorithms.

lin	accel	single	finite	sat	sgn
0.37	0.31	0.38	0.75	0.38	1.18

a determining factor on the stability of the grid [42]. The solutions of the existing linear methods [6,46,65] may assign any rate of change $\dot{x}_i$ to the power generators. Thus, they cannot address RRLs, which, in reality, the solutions cannot be followed by the generators. This may fail the feasibility and lead to sub-optimality. In contrast, considering the nonlinearity $g(\cdot)$ as saturation function (18), the proposed nonlinear solutions in this work can address RRL constraints where the limits can be tuned by the parameter κ. A comparing example is given next.

7.1. Allocation with no delay

Assume $n = 50$ generators with the supply–demand constraint $D = 3200$ MW. Initially, it is considered that $x_i = \frac{D}{n} = 64$ MW at every generator i. The parameters in (54) are randomly set from Table 2 using MATLAB `randi` function. For power allocation, as an academic example, we consider protocol (25) with saturated nonlinearity $g_\kappa(\cdot)$ with level $\kappa = \frac{1}{60}$ on the node dynamics and protocol (26) with $g_f(\cdot)$ as the sign-based function (19) (with $v_1 = 0.4$, $v_2 = 1.6$) on the links. The network is considered as a random Erdos–Renyi graph (with link probability $p = 0.2$) with random symmetric link weights $0.005 \leq W_{ij} \leq 0.025$. We provide a comparative simulation analysis to support our results. Both solutions are compared with some state-of-the-art solutions in the literature in Fig. 2 for $T = 1$; namely: linear [46], accelerated linear [66], finite-time [26], and single-bit [25] solutions. For the single-bit protocol [25] we decreased the link weights by 80% to reduce the chattering effect in Fig. 3. Note that the RRL constraint is only met by our saturated solution as shown in Fig. 3. In other words, in a real scenario the generators cannot follow the iterative solution by [25,26,46,66] since their power generation rates at some intervals violate the RRLs. This may either cause feasibility gap or optimality gap in real world. But our saturated-based solution admits the RRLs. The min RRL value equal to 1 MW/min (or $\frac{1}{60}$ MW/sec) is considered that meets the requirement by all the generators.[12]

Box constraints $m_i \leq x_i \leq M_i$ on the generators are considered by adding smooth penalty terms with $\sigma = 2$. We modify the objective as $f_i^\sigma = f_i(x_i) + c([x_i - M_i]^+)^2 + c([m_i - x_i]^+)^2$ with $c = 1$. The evolution of power states under the sign-based dynamics is shown in Fig. 4, where both feasibility and power limit constraints are met. Note that our given assumptions ensure that at every time step the deviated power at nodes on two sides of every link is balanced such that the feasibility constraint is satisfied (i.e., the generated power equals the demand) at all times.

To compare the computational complexity of the algorithms, the elapsed times (for one iteration) are given in Table 3. The simulation is done with MATLAB R2021b Intel Core i5 @ 2.4 GHz processor RAM 8 GB.

Fig. 2. The performance of our node-based protocol (25) subject to saturation and link-based protocol (26) with sign-based nonlinearity compared with linear [46], accelerated linear [66], finite-time [26], and single-bit [25] protocols.

Fig. 3. The rate of generated power $\dot{x}_i$ (in MW/sec) at a sample generator $i =$ are compared under different solutions. Only our proposed saturated nonlinear model admits the RRL constraints on the generator dynamics (shown by dashed lines).

Fig. 4. The allocated power states x_i (in MW) under sign-based dynamics and box constraints on the output powers as $m_i \leq x_i \leq M_i$. The feasibility is checked by the average of output powers which is constant (black dashed line). The normalized allocated power states x_i are also shown to verify the box constraints.

---

[12] Even though we used the same min RRL in this simulation, this parameter can be further tuned for different generators in dynamics (14)–(15) via the link weights and the node degrees.
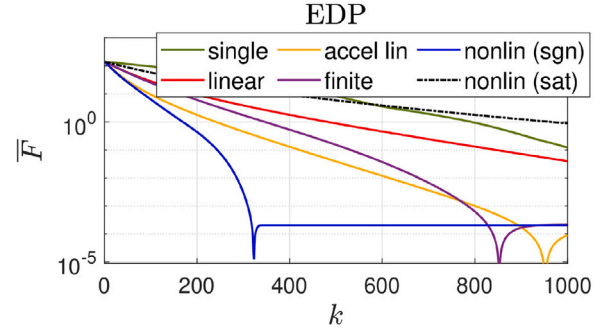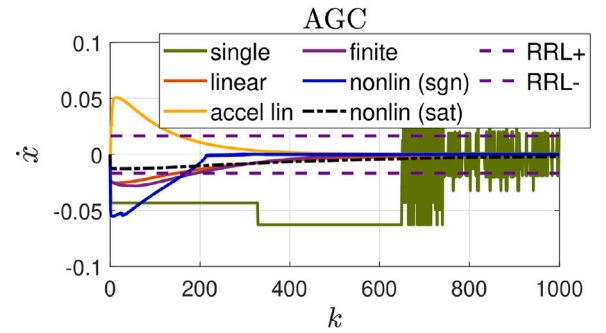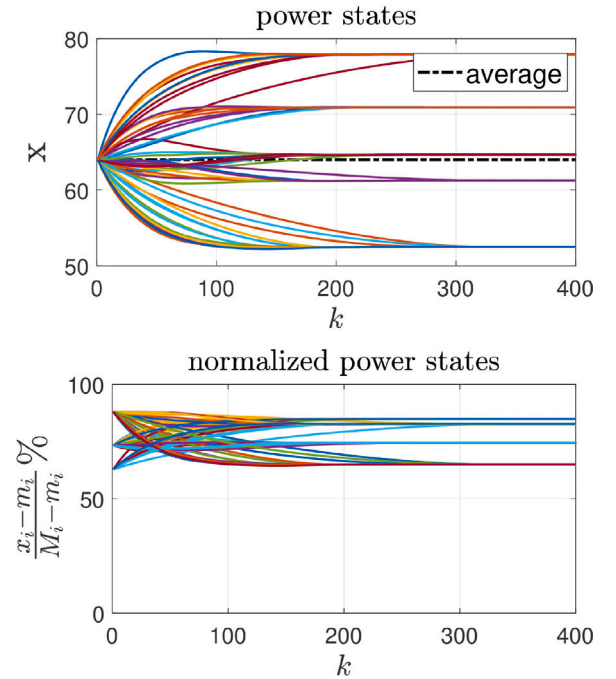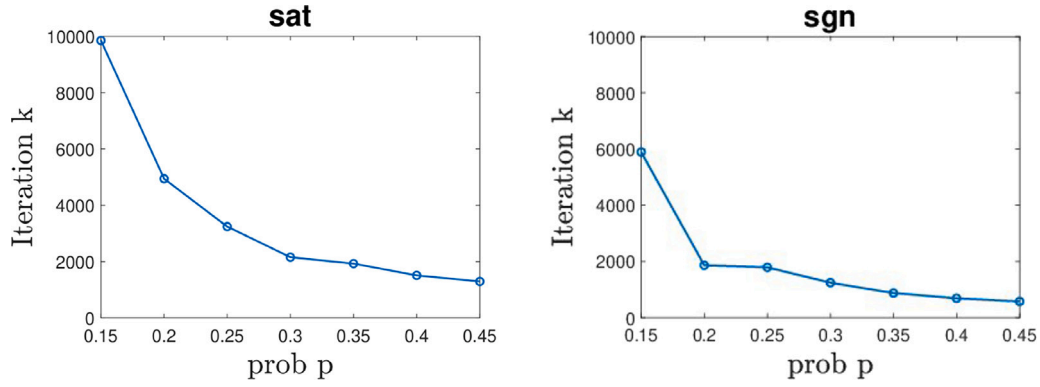
Fig. 5. The number of iterations to reach residual $\overline{F} = 10^{-2}$ versus ER network connectivity tuned by the link probability p for sat (level $\kappa = 1$) and sgn ($v_1 = 0.4$, $v_2 = 1.6$) nonlinearity.
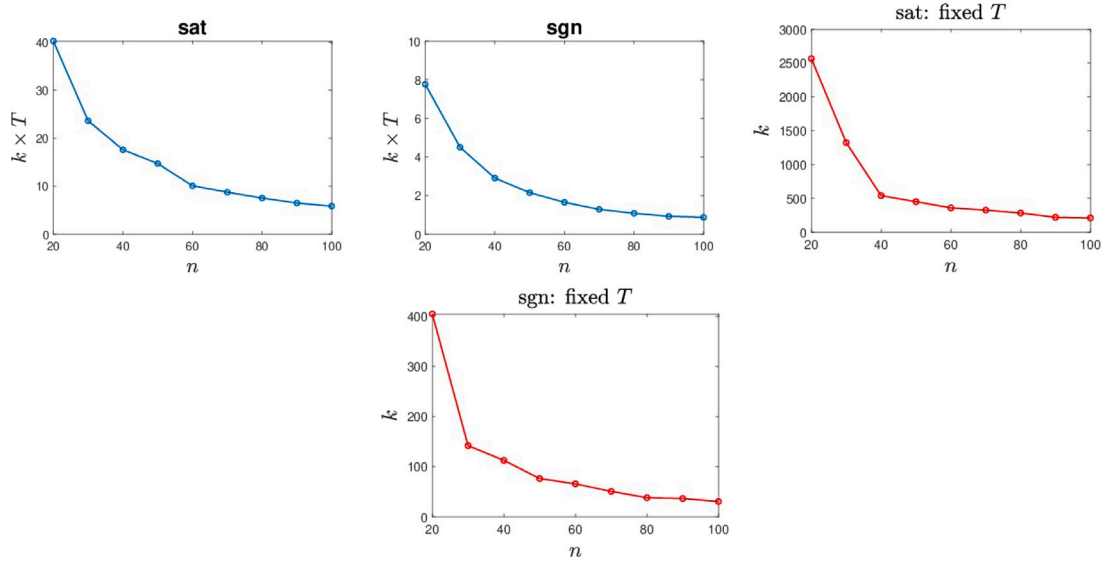

Fig. 6. The convergence time/iterations to reach residual $\overline{F} = 0.01$ versus ER network size n for sat (level $\kappa = 2$) and sgn ($v_1 = 0.5$, $v_2 = 1.5$) nonlinearity. (Top) diminishing step-sizes as size increases, (Bottom) fixed step-sizes.

Table 4
The number of iterations k to reach $\overline{F} = 1$.

lin	accel	single	finite	sat	sgn
480	239	791	345	962	168

In Table 4, we compared the number of iterations to reach (a predefined) cost residual $\overline{F} = 1$. For example, within this range around the optimal value the solution is considered good enough. Note that our proposed sign-based solution, as compared to other solutions, although having more computational complexity converges very fast and within fewer number of iterations.

Next, we study the network density on the number of iterations to reach a certain residual (as the termination criteria of the algorithm). In Fig. 5, this termination point is $\overline{F} = 0.01$ and the ER link probability is changed from 15% to 45%. The link weights W_{ij} are chosen randomly in the range [0.02 0.12]. The simulation is averaged over 5 Monte-Carlo (MC) trials with $T = 0.05$.

The convergence time/iterations to reach $\overline{F} = 10^{-2}$ versus network size n is shown in Fig. 6. The parameters are as follows: $p = 30\%$ as the ER link probability, $W_{ij} \in [0.05\ 0.2]$, and 5 MC trials. Two different scenarios are considered for T: diminishing by size as $T = 0.05(1 - \frac{n-20}{100})$ and fixed step size $T = 0.025$.

In Fig. 7, for the same termination criteria $\overline{F} = 10^{-2}$, the number of iterations for different step sizes T and link densities p are given. As we see in the figure, the convergence iteration multiplied by the step size is almost constant irrespective of the network connectivity (and eigen-spectrum) and the step size T. Note that the T values are chosen small enough for convergence, and for large values of T violating Eq. (44) the solution may not necessarily converge.

7.2. Allocation with delayed information-exchange

Next, we consider a cyclic network of $n = 5$ generators under RRL limits $\kappa = \frac{1}{60}$ MW/sec in the presence of time delays. The data in Table 2 is used which resembles IEEE 14-Bus test system [41]. The time-evolution of the generated powers and the Lyapunov function $\overline{F}$ (the residual) are simulated for random parameters. For the quadratic cost (54), $u = \max\{\gamma_i\} = 0.04$. We have $\varepsilon = 0.0166$, $K_g = 1$, $\lambda_2 = 1.38$, $T = 1$ and $\lambda_n = 3.61$. Using Eq. (44), for any $T < 0.045$ the solution converges in the absence of delays. This is a sufficient bound and to some extent not very tight.

Case I: In this case, using (47), we update the generator states at every $\overline{\tau}$ steps as in Section 6.1. Following Remark 9, we consider information exchange over the longer time-scale $\overline{k}$. The time-evolution of the Lyapunov function $\overline{F}$ (representing the residual) for some values of $\overline{\tau}$ is given in Fig. 8. Clearly, the convergence rate decreases with the increase in the time delays.
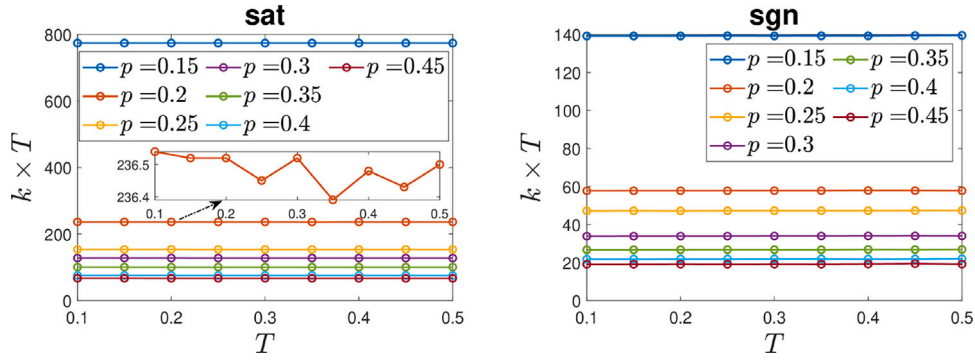
Fig. 7. The convergence iteration×T to reach residual $\overline{F} = 10^{-2}$ versus step size T for sat (level $\kappa = 3$) and sgn ($v_1 = 0.5$, $v_2 = 1.5$) nonlinearity. The connectivity of the ER network is changed via the link density p.
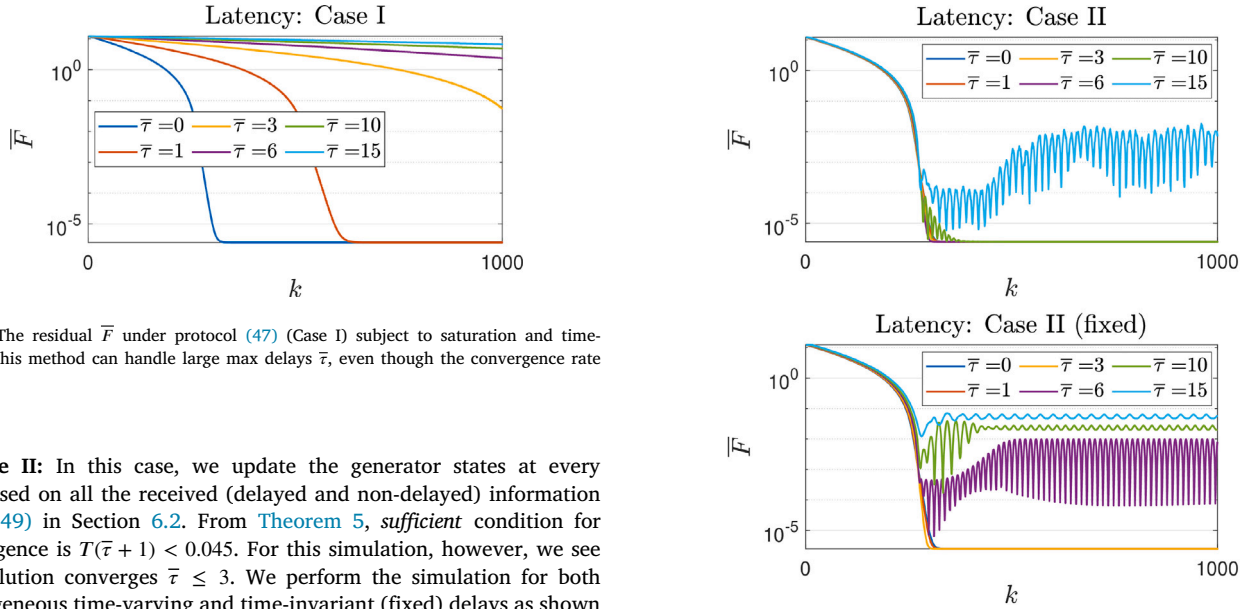

Fig. 8. The residual $\overline{F}$ under protocol (47) (Case I) subject to saturation and time-delays. This method can handle large max delays $\overline{\tau}$, even though the convergence rate is low.

Case II: In this case, we update the generator states at every step based on all the received (delayed and non-delayed) information using (49) in Section 6.2. From Theorem 5, *sufficient* condition for convergence is $T(\overline{\tau} + 1) < 0.045$. For this simulation, however, we see that solution converges $\overline{\tau} \leq 3$. We perform the simulation for both heterogeneous time-varying and time-invariant (fixed) delays as shown in Fig. 9.

8. Simulation of non-smooth quantized optimization over uniformly connected networks

In this section, we run the simulations over a random *dynamic* network of $n = 100$ agents. The network is not connected at any time, while it is B-connected over every $B = 100$ iterations (Assumption 2), i.e., $\bigcup_k^{k+100} \mathcal{G}(k)$ is connected. We consider strongly-convex logarithmic functions $f_i(\cdot)$ defined as [46],

$$f_i(x_i) = \frac{1}{2}\alpha_i(x_i - \gamma_i)^2 + \zeta \log(1 + \exp(\beta_i(x_i - \eta_i))), \quad (55)$$

with the coefficients randomly chosen as $0 < \alpha_i < 0.2$, $-0.2 < \beta_i < 0.2$, $-0.3 < \gamma_i < 0.3$, $0 < \eta_i < 0.6$, $\zeta = 0.2$, and random constraint parameters in problem (2) chosen as $-2 < a_i < 2$ and $b = 10$. To optimize this objective we apply (26) (with $T = 0.1$) under logarithmic quantization $g_l(\cdot)$ in Eq. (17). This locally Lipschitz function satisfies $\left(1 - \frac{\delta}{2}\right)z \leq \frac{g_l(z)}{z} \leq \left(1 + \frac{\delta}{2}\right)z$ and, thus, Assumption 4 holds. We considered a composition of dynamics (25)–(26) via $g_\kappa\left(g_l(\partial f_i) - g_l(\partial f_j)\right)$ with $\kappa = 1$. The time-evolution of the residual $\overline{F}$ for different quantization levels δ is given in Fig. 10. We further compare the performance under different time-delay models by assigning random heterogeneous delays to the links. For Case II, two scenarios are given (i) time-varying, and (ii) time-invariant (fixed) delays. For Case I, from Remark 9, we consider updating over time-scale $\overline{k}$ using composition of dynamics (46)–(47). Simulations are shown in Fig. 11 with parameters: $\delta = 0.125$, $T =$

Fig. 9. The residual $\overline{F}$ under DT protocol (49) under Case II with (top) time-varying and (bottom) time-invariant delays. For small $\overline{\tau}$ the convergence is faster than Case I, but for large $\overline{\tau}$ this approach may not converge (e.g., $\overline{\tau} = 15$).

0.05, $\overline{\tau} = 2, 6$. As shown in Fig. 11, the solution by Case II does not necessarily converge while the solution by Case I converges. This simulation shows that for small $\overline{\tau}$ (satisfying Theorem 5) Case II leads to faster convergence. On the other hand, for larger $\overline{\tau}$ Case I is a better delay-tolerant mechanism.

Note that the notion of information quantization and time-delays although prevalent in real-world networked systems cannot be addressed by the existing primal-based [4,6,14,33,46,47,50,65–68] and dual-based solutions [34–40]. In other words, these existing literature assume ideal network condition and there is no guarantee that their solution converge under quantized information and/or data transmission delays.

9. Conclusion and future directions

This paper proposes anytime-feasible (Laplacian-gradient) solutions subject to model nonlinearities to solve distributed sum-preserving resource allocation and coupling-constraint optimization over uniformly-connected networks (not necessarily connected at all times). The convergence to the optimal value is proved for general strongly sign-preserving nonlinearities. In addition, two scenarios are proposed to overcome heterogeneous delays over the network. For large delays and
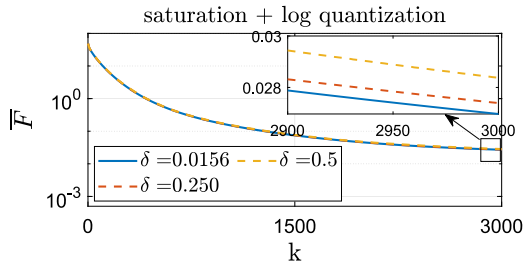
Fig. 10. This figure shows the residual $\overline{F}$ for quantized communications and saturated actuation (composition of nonlinear protocols (25)–(26)) over a dynamic uniformly-connected network. δ denotes the quantization level.
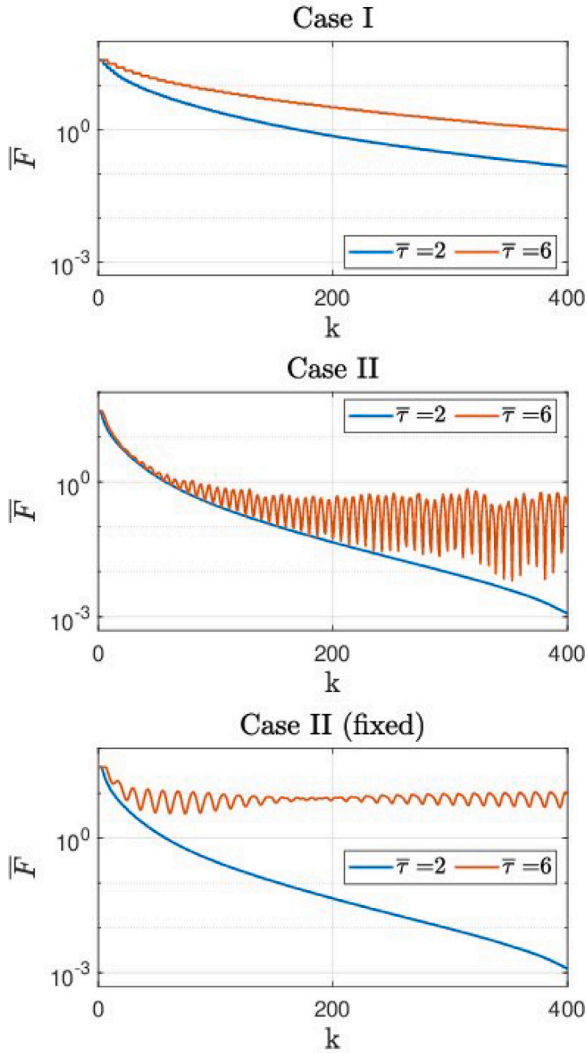

Fig. 11. This figure shows the time-evolution of the residual $\overline{F}$ subject to delays $\overline{\tau} = 2, 6$ and logarithmic quantization. For Case II both time-varying and time-invariant (fixed at every link) delays are considered. For the case of a smaller delay value, the solution by Case II converges faster, while, for the larger delay value $\overline{\tau} = 6$, the solution by Case I gives better performance.

low buffers, we proposed to update the agents' states over a longer time scale after receiving (at least) 1 delayed packet over every link. On the other hand, faster convergence can be achieved, for smaller time delays, by updating at the same time scale of the communication and

using all the received (possibly) delayed packets at each iteration. The results are given for undirected and balanced graphs with bounded and link-symmetric delays.

Allocation strategies over lossy networks with link failure or packet drop are another direction of our current research. As future research directions, application to asynchronous scheduling under quantized dynamics [78], robust (and noise-resilient) sign-based [11,28], or single-bit [25] dynamics are of interest. Recall that for non "strongly" sign-preserving solutions, the ε-accuracy needs to be addressed to give an estimate of the optimality gap. Another future research direction is to extend the results to non-convex problems, which is a bottleneck and more interesting for the industry.

CRediT authorship contribution statement

Mohammadreza Doostmohammadian: Wrote the paper, Done the simulations, Discussed the idea and conceptualization, Read and revised the paper. **Alireza Aghasi:** Discussed the idea and conceptualization, Read and revised the paper. **Maria Vrakopoulou:** Discussed the idea and conceptualization, Read and revised the paper. **Hamid R. Rabiee:** Discussed the idea and conceptualization, Read and revised the paper. **Usman A. Khan:** Discussed the idea and conceptualization, Read and revised the paper. **Themistoklis Charalambous:** Discussed the idea and conceptualization, Read and revised the paper.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Mohammadreza doostmohammadian reports was provided by Semnan University. Mohammadreza doostmohammadian reports a relationship with Semnan University that includes: employment. Mohammadreza doostmohammadian has patent pending to mohammadreza doostmohammadian. none

Data availability

No data was used for the research described in the article.

References

[1] M. Doostmohammadian, A. Aghasi, T. Charalambous, U.A. Khan, Distributed support vector machine over dynamic balanced directed networks, IEEE Control Syst. Lett. 6 (2021) 758–763.

[2] U.A. Khan, W.U. Bajwa, A. Nedić, M.G. Rabbat, A.H. Sayed, Optimization for data-driven learning and control, Proc. IEEE 108 (11) (2020) 1863–1868.

[3] D.K. Molzahn, F. Dörfler, H. Sandberg, S.H. Low, S. Chakrabarti, R. Baldick, J. Lavaei, A survey of distributed optimization and control algorithms for electric power systems, IEEE Trans. Smart Grid 8 (6) (2017) 2941–2962.

[4] S. Yang, S. Tan, J. Xu, Consensus based approach for economic dispatch problem in a smart grid, IEEE Trans. Power Syst. 28 (4) (2013) 4416–4426.

[5] C.N. Hadjicostis, T. Charalambous, Asynchronous coordination of distributed energy resources for the provisioning of ancillary services, in: 49th IEEE Allerton Conference on Communication, Control, and Computing, 2011, pp. 1500–1507.

[6] A. Cherukuri, J. Cortés, Distributed generator coordination for initialization and anytime optimization in economic dispatch, IEEE Trans. Control Netw. Syst. 2 (3) (2015) 226–237.

[7] J. Wei, X. Yi, H. Sandberg, K.H. Johansson, Nonlinear consensus protocols with applications to quantized communication and actuation, IEEE Trans. Control Netw. Syst. 6 (2) (2019) 598–608.

[8] C.N. Hadjicostis, N.H. Vaidya, A.D. Domínguez-García, Robust distributed average consensus via exchange of running sums, IEEE Trans. Automat. Control 61 (6) (2016) 1492–1507.

[9] S.E. Parsegov, A.E. Polyakov, P.S. Shcherbakov, Fixed-time consensus algorithm for multi-agent systems with integrator dynamics, IFAC Proc. Vol. 46 (27) (2013) 110–115.

[10] K. Garg, M. Baranwal, A.O. Hero, D. Panagou, Fixed-time distributed optimization under time-varying communication topology, 2019, arXiv preprint arXiv:1905.10472.

[11] S.S. Stanković, M. Beko, M.S. Stanković, Robust nonlinear consensus seeking, in: 58th IEEE Conference on Decision and Control, 2019, pp. 4465–4470.

[12] A. Agarwal, J.C. Duchi, Distributed delayed stochastic optimization, in: 51st IEEE Conference on Decision and Control, 2012, pp. 5451–5452.

[13] H. Al-Lawati, S.C. Draper, Gradient delay analysis in asynchronous distributed optimization, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2020, pp. 4207–4211.

[14] D. Wang, Z. Wang, M. Chen, W. Wang, Distributed optimization for multi-agent systems with constraints set and communication time-delay over a directed graph, Inform. Sci. 438 (2018) 1–14.

[15] X. Wang, Y. Hong, X. Sun, K. Liu, Distributed optimization for resource allocation problems under large delays, IEEE Trans. Ind. Electron. 66 (12) (2019) 9448–9457.

[16] C. Hu, G. Wen, S. Wang, J. Fu, W. Yu, Distributed multiagent reinforcement learning with action networks for dynamic economic dispatch, IEEE Trans. Neural Netw. Learn. Syst. (2023).

[17] P. Dai, W. Yu, G. Wen, S. Baldi, Distributed reinforcement learning algorithm for dynamic economic dispatch with unknown generation cost functions, IEEE Trans. Ind. Inform. 16 (4) (2019) 2258–2267.

[18] M. Doostmohammadian, A. Aghasi, M. Pirani, E. Nekouei, U.A. Khan, T. Charalambous, Fast-convergent anytime-feasible dynamics for distributed allocation of resources over switching sparse networks with quantized communication links, in: European Control Conference, 2022, pp. 84–89.

[19] A. Nedic, A. Olshevsky, A. Ozdaglar, J.N. Tsitsiklis, Distributed subgradient methods and quantization effects, in: 47th IEEE Conference on Decision and Control, 2008, pp. 4177–4184.

[20] A.I. Rikos, C.N. Hadjicostis, Distributed average consensus under quantized communication via event-triggered mass splitting, IFAC-PapersOnLine 53 (2) (2020) 2957–2962.

[21] A.I. Rikos, T. Charalambous, K.H. Johansson, C.N. Hadjicostis, Privacy-preserving event-triggered quantized average consensus, in: 59th IEEE Conference on Decision and Control, 2020, pp. 6246–6253.

[22] E. Nekouei, T. Alpcan, G.N. Nair, R.J. Evans, Convergence analysis of quantized primal-dual algorithms in network utility maximization problems, IEEE Trans. Control Netw. Syst. 5 (1) (2016) 284–297.

[23] E. Nekouei, G.N. Nair, T. Alpcan, Performance analysis of gradient-based nash seeking algorithms under quantization, IEEE Trans. Automat. Control 61 (12) (2016) 3771–3783.

[24] Z. Liu, A. Saberi, A.A. Stoorvogel, D. Nojavanzadeh, Global regulated state synchronization for homogeneous networks of non-introspective agents in presence of input saturation: Scale-free nonlinear and linear protocol designs, Automatica 119 (2020) 109041.

[25] M. Doostmohammadian, Single-bit consensus with finite-time convergence: Theory and applications, IEEE Trans. Aerosp. Electron. Syst. 56 (4) (2020) 3332–3338.

[26] G. Chen, J. Ren, E.N. Feng, Distributed finite-time economic dispatch of a network of energy resources, IEEE Trans. Smart Grid 8 (2) (2016) 822–832.

[27] B. Ning, Q. Han, Z. Zuo, Distributed optimization for multiagent systems: An edge-based fixed-time consensus approach, IEEE Trans. Cybern. 49 (1) (2017) 122–132.

[28] J. Wei, A.R.F. Everts, M.K. Camlibel, A.J. van der Schaft, Consensus dynamics with arbitrary sign-preserving nonlinearities, Automatica 83 (2017) 226–233.

[29] X. Wu, S. Magnusson, M. Johansson, A new family of feasible methods for distributed resource allocation, in: 60th IEEE Conference on Decision and Control, 2021, pp. 3355–3360.

[30] A.I. Rikos, A. Grammenos, E. Kalyvianaki, C.N. Hadjicostis, T. Charalambous, K.H. Johansson, Optimal CPU scheduling in data centers via a finite-time distributed quantized coordination mechanism, in: 60th IEEE Conference on Decision and Control, 2021, pp. 6276–6281.

[31] Y. Zhu, W. Ren, W. Yu, G. Wen, Distributed resource allocation over directed graphs via continuous-time algorithms, IEEE Trans. Syst. Man Cybern. Syst. 51 (2) (2019) 1097–1106.

[32] N.S. Aybat, E. Yazdandoost Hamedani, Distributed primal-dual method for multi-agent sharing problem with conic constraints, in: 50th IEEE Asilomar Conference on Signals, Systems and Computers, 2016, pp. 777–782.

[33] A. Nedić, A. Olshevsky, W. Shi, Improved convergence rates for distributed resource allocation, in: IEEE Conference on Decision and Control, IEEE, 2018, pp. 172–177.

[34] G. Banjac, F. Rey, P. Goulart, J. Lygeros, Decentralized resource allocation via dual consensus ADMM, in: IEEE American Control Conference, 2019, pp. 2789–2794.

[35] A. Falsone, I. Notarnicola, G. Notarstefano, M. Prandini, Tracking-ADMM for distributed constraint-coupled optimization, Automatica 117 (2020) 108962.

[36] T. Chang, A proximal dual consensus ADMM method for multi-agent constrained optimization, IEEE Trans. Signal Process. 64 (14) (2016) 3719–3734.

[37] W. Jiang, M. Doostmohammadian, T. Charalambous, Distributed resource allocation via ADMM over digraphs, in: IEEE 61st Conference on Decision and Control, CDC, IEEE, 2022, pp. 5645–5651.

[38] A. Falsone, K. Margellos, M. Prandini, A distributed iterative algorithm for multi-agent MILPs: Finite-time feasibility and performance characterization, IEEE Control Syst. Lett. 2 (4) (2018) 563–568.

[39] N. Aybat, E. Yazdandoost Hamedani, A distributed ADMM-like method for resource sharing over time-varying networks, SIAM J. Optim. 29 (4) (2019) 3036–3068.

[40] M. Doostmohammadian, W. Jiang, T. Charalambous, DTAC-ADMM: Delay-tolerant augmented consensus ADMM-based algorithm for distributed resource allocation, in: IEEE 61st Conference on Decision and Control, CDC, IEEE, 2022, pp. 308–315.

[41] S. Kar, G. Hug, Distributed robust economic dispatch in power systems: A consensus + innovations approach, in: IEEE Power and Energy Society General Meeting, 2012, pp. 1–8.

[42] A.N. Venkat, I.A. Hiskens, J.B. Rawlings, S.J. Wright, Distributed MPC strategies with application to power system automatic generation control, IEEE Trans. Control Syst. Technol. 16 (6) (2008) 1192–1206.

[43] M. Doostmohammadian, A. Aghai, A.I. Rikos, A. Grammenos, E. Kalyvianaki, C.N. Hadjicostis, K.H. Johansson, T. Charalambous, Distributed CPU scheduling subject to nonlinear constraints, in: IEEE Conference on Control Technology and Applications, pp. 746–751.

[44] H. Sayyaadi, M. Moarref, A distributed algorithm for proportional task allocation in networks of mobile agents, IEEE Trans. Automat. (ISSN: 0018-9286) 56 (2) (2011) 405–410, http://dx.doi.org/10.1109/TAC.2010.2089653.

[45] A. Falsone, K. Margellos, S. Garatti, M. Prandini, Dual decomposition for multi-agent distributed optimization with coupling constraints, Automatica 84 (2017) 149–158.

[46] L. Xiao, S. Boyd, Optimal scaling of a gradient method for distributed resource allocation, J. Optim. Theory Appl. 129 (3) (2006) 469–488.

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

[48] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, 2009.

[49] M. Doostmohammadian, A. Aghasi, M. Vrakopoulou, T. Charalambous, 1St-order dynamics on nonlinear agents for resource allocation over uniformly-connected networks, in: IEEE Conference on Control Technology and Applications, 2022, pp. 1184–1189.

[50] H. Lakshmanan, D.P. De Farias, Decentralized resource allocation in dynamic networks of agents, SIAM J. Optim. 19 (2) (2008) 911–940.

[51] D.P. Bertsekas, Necessary and sufficient conditions for a penalty method to be exact, Math. Progr. 9 (1) (1975) 87–99.

[52] D.P. Bertsekas, A. Nedic, A.E. Ozdaglar, Convexity, duality, and lagrange multipliers, in: Lecture Notes, MIT Press, 2001.

[53] A. Nedić, A. Olshevsky, Distributed optimization over time-varying directed graphs, IEEE Trans. Automat. Control 60 (3) (2014) 601–615.

[54] W. Ren, R.W. Beard, Consensus seeking in multiagent systems under dynamically changing interaction topologies, IEEE Trans. Automat. Control 50 (5) (2005) 655–661.

[55] N.H. Vaidya, C.N. Hadjicostis, A.D. Dominguez-Garcia, Robust average consensus over packet dropping links: Analysis via coefficients of ergodicity, in: IEEE 51st IEEE Conference on Decision and Control, 2012, pp. 2761–2766.

[56] R. Olfati-Saber, R.M. Murray, Consensus problems in networks of agents with switching topology and time-delays, IEEE Trans. Automat. Control 49 (9) (2004) 1520–1533.

[57] J.L. Gross, J. Yellen, Handbook of Graph Theory, CRC Press, 2004.

[58] H. Wang, P. Van Mieghem, Algebraic connectivity optimization via link addition, in: 3rd International Conference on Bio-Inspired Models of Network, Information and Computing Systems, 2008, pp. 1–8.

[59] C.N. Hadjicostis, T. Charalambous, Average consensus in the presence of delays in directed graph topologies, IEEE Trans. Automat. Control 59 (3) (2014) 763–768.

[60] T. Charalambous, Y. Yuan, T. Yang, W. Pan, C.N. Hadjicostis, M. Johansson, Distributed finite-time average consensus in digraphs in the presence of time delays, IEEE Trans. Control Netw. Syst. 2 (4) (2015) 370–381.

[61] T. Charalambous, C.N. Hadjicostis, Average consensus in the presence of dynamically changing directed topologies and time delays, in: 53rd IEEE Conference on Decision and Control, 2014, pp. 709–714.

[62] K.I. Tsianos, M.G. Rabbat, Distributed consensus and optimization under communication delays, in: 49th Annual Allerton Conference on Communication, Control, and Computing, 2011, pp. 974–982.

[63] M. Maróti, B. Kusy, G. Simon, A. Lédeczi, The flooding time synchronization protocol, in: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, 2004, pp. 39–49.

[64] R.A. Berry, R.G. Gallager, Communication over fading channels with delay constraints, IEEE Trans. Inform. Theory 48 (5) (2002) 1135–1149.

[65] B. Gharesifard, J. Cortés, Distributed continuous-time convex optimization on weight-balanced digraphs, IEEE Trans. Automat. Control 59 (3) (2013) 781–786.

[66] E. Ghadimi, M. Johansson, I. Shames, Accelerated gradient methods for networked optimization, in: IEEE American Control Conference, 2011, pp. 1668–1673.

[67] T.T. Doan, A. Olshevsky, Distributed resource allocation on dynamic networks in quadratic time, Systems Control Lett. 99 (2017) 57–63.

[68] T.T. Doan, C.L. Beck, Distributed Lagrangian methods for network resource allocation, in: IEEE Conference on Control Technology and Applications, CCTA, 2017, pp. 650–655.

[69] A. Kashyap, T. Basar, R. Srikant, Consensus with quantized information updates, in: 45th IEEE Conference on Decision and Control, 2006, pp. 2728–2733.

[70] T. Binazadeh, M. Bahmani, Design of robust controller for a class of uncertain discrete-time systems subject to actuator saturation, IEEE Trans. Automat. Control 62 (3) (2016) 1505–1510.

[71] Y. Wang, Y. Song, D.J. Hill, M. Krstic, Prescribed-time consensus and containment control of networked multiagent systems, IEEE Trans. Cybern. 49 (4) (2019) 1138–1147.

[72] T. Charalambous, C.N. Hadjicostis, M. Johansson, Distributed minimum-time weight balancing over digraphs, in: 6th International Symposium on Communications, Control and Signal Processing, 2014, pp. 190–193.

[73] D. Jurafsky, J.H. Martin, Speech and Language Processing, Prentice Hall, 2020.

[74] Y. Nesterov, Introductory lectures on convex programming, Volume I: Basic course, Lect. Notes 3 (4) (1998) 5.

[75] N. Amjady, H. Nasiri-Rad, Economic dispatch using an efficient real-coded genetic algorithm, IET Gener. Trans. Distrib. 3 (3) (2009) 266–278.

[76] A.J. Wood, B.F. Wollenberg, G.B. Sheble, Power Generation Operation and Control, John Wiley & Sons, 1996.

[77] Reliability test system grid modernization lab consortium data, 2022, https://github.com/GridMod/RTS-GMLC. Online; (Accessed 10 August 2022).

[78] A. Grammenos, T. Charalambous, E. Kalyvianaki, CPU scheduling in data centers using asynchronous finite-time distributed coordination mechanisms, IEEE Trans. Netw. Sci. Eng. (2023).