Received 11 May 2022; revised 17 August 2022; accepted 24 September 2022. Date of publication 28 September 2022; date of current version 13 October 2022. Recommended by Senior Editor Prof. Lacra Pavel.

Digital Object Identifier 10.1109/OJCSYS.2022.3210453

Distributed Anytime-Feasible Resource Allocation Subject to Heterogeneous Time-Varying Delays

MOHAMMADREZA DOOSTMOHAMMADIAN [1], ALIREZA AGHASI [2], APOSTOLOS I. RIKOS [3], ANDREAS GRAMMENOS [4,5], EVANGELIA KALYVIANAKI [5], CHRISTOFOROS N. HADJICOSTIS [6] (Fellow, IEEE), KARL H. JOHANSSON [3] (Fellow, IEEE), AND THEMISTOKLIS CHARALAMBOUS [1,6] (Senior Member, IEEE)

[1]Department of Electrical Engineering and Automation, School of Electrical Engineering, Aalto University, Espoo, Finland
[2]Robinson College of Business, Georgia State University, Atlanta, GA USA
[3]Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm, Sweden
[4]The Alan Turing Institute, London, U.K.
[5]Department of Computer Science and Technology, University of Cambridge, Cambridge, U.K.
[6]Department of Electrical and Computer Engineering, School of Engineering, University of Cyprus, Nicosia, Cyprus

CORRESPONDING AUTHOR: MOHAMMADREZA DOOSTMOHAMMADIAN (e-mail: doost@semnan.ac.ir).

This work was supported by the European Commission through the H2020 Project FinEst Twins under Agreement 856602.

ABSTRACT This paper considers distributed allocation strategies, formulated as a distributed sum-preserving (fixed-sum) allocation of resources over a multi-agent network in the presence of heterogeneous arbitrary time-varying delays. We propose a double time-scale scenario for unknown delays and a faster single time-scale scenario for known delays. Further, the links among the nodes are considered subject to certain nonlinearities (e.g, quantization and saturation/clipping). We discuss different models for nonlinearities and how they may affect the convergence, sum-preserving feasibility constraint, and solution optimality over general weight-balanced uniformly strongly connected networks and, further, time-delayed undirected networks. Our proposed scheme works in a variety of applications with general non-quadratic strongly-convex smooth objective functions. The non-quadratic part, for example, can be due to additive convex penalty or barrier functions to address the local box constraints. The network can change over time, is not necessarily connected at all times, but is only assumed to be uniformly-connected. The novelty of this work is to address all-time feasible Laplacian gradient solutions in presence of nonlinearities, switching digraph topology (not necessarily all-time connected), and heterogeneous time-varying delays.

INDEX TERMS Allocation strategies, balanced digraphs, networked constrained optimization, sum-preserving coupling-constraint.

I. INTRODUCTION

Resource allocation strategies in the real world are subject to possible nonlinear constraints, e.g., quantization [1], [2] over a finite number of bits and the so-called *clipping*. Some works even consider *single-bit* data-exchange to reduce communication loads over the network [3], [4]. Such nonlinearities, in general, may affect the performance of the distributed algorithms. The existing distributed optimization methods in both constrained [5] and unconstrained settings [6], [7], [8] are mostly linear. This paper proposes a discrete-time algorithm for distributed resource allocation and constrained optimization subject to link nonlinearity and heterogeneous time delays. The problem is to optimally assign (a fixed amount of) resources to minimize (maximize) a cost (utility) function, with applications from coverage in deployment problems over robotic networks [9] to generator coordination for economic dispatch [10] over the energy grid, and even network epidemics [11]. Existing literature assume ideal (linear and unconstrained) communications, i.e., the exchanged information over a network can be of any real value. This is

not practical as communication links are digital and of limited bandwidth. In addition, communication links may experience delays (e.g., due to packet re-transmission). Under limited network bandwidth and/or latency, the existing solutions may not be optimal or may lose feasibility. The feasibility constraint ensures the balance between assigned resources and the overall demand. The proposed strategies can be used for resource management over Cloud infrastructure (as in [1]) while considering heterogeneous delays over links, local box constraints on the states, and quantized (discrete-valued) communications among the servers.

A. RELATED LITERATURE

The classical work in the context of constrained distributed optimization mainly assume linear and ideal communication and data transmission, e.g., see the seminal work by [12] which considers a Laplacian-based constraint on the states. Some recent works consider unconstrained distributed optimization [13] and consensus optimization [14] subject to communication delays, resource allocation over open networks subject to arrivals/departures of nodes [15], and double averaging and projection-based solutions over static digraphs [16]. Bit allocation for distributed optimization setups is also considered in [17]. In resource allocation, solution feasibility is crucial for the resource-demand balance (at the termination point of the algorithm) to avoid service disruption and even system breakdown [18]. The Laplacian gradient solutions benefit from *anytime feasibility* [10], [18], i.e., the sum-preserving equality-constraint holds at all iterations of the algorithm, in contrast to *asymptotic* feasibility in the primal-dual and ADMM-based solutions [19], [20], [21], [22], [23]. Some other concerns in distributed resource allocation are: (i) uniform-connectivity in case of dynamic and sparse mobile networks in contrast to all-time connectivity [19], [20], [21], [22], [23]. (ii) Latency over the network to account for possible time-varying heterogeneous delays due to data exchanges over unreliable communication links between agents or even asynchronicity between the nodes. The time-delays may even cause *feasibility gap* in the allocation algorithm [24]. Finally, (iii) possible nonlinearities in the model mainly due to quantization [1] and discrete-value optimization [2] in contrast to the ideal linear models.

Some example resource-allocation applications include economic dispatch and generation control over the smart-grid [10], [25], [26], [27] and CPU scheduling over the network of data centers [1]. Apart from these *quadratic* models, some other works also address possible *non-quadratic* objectives [28], [29], [30] to span more application scenarios. In general, nonlinear dynamics (either inherent to the system model or additive constraints due to limited capacity/storage) are prevalent in practical applications and cannot be addressed with the existing linear algorithms. Some examples are: the ramp-rate-limit on the generators' dynamics for automatic generation control [27], impulsive-noise resiliency in consensus algorithms [31], or convergence in finite/fixed-time [4].

Quantization or clipping (and general strongly sign-preserving odd nonlinearities) over dynamic networks while satisfying distributed anytime-feasibility in presence of (possible) time-delays are not addressed in the existing optimization works (to our best knowledge). For example, the work [24] addresses homogeneous time-delays at all links with some *feasibility-gap* over switching undirected (all-time) connected graphs. Possible local box constraints on the states may add even more complexity to the model [18]. Recall that some of the mentioned model constraints are discussed in consensus literature [2], [3], [4], [31], [32], [33], but not well-addressed in their general form in the networked optimisation research and this paper fills the gap.

B. MAIN CONTRIBUTIONS

The proposed distributed allocation protocol in this work is (i) anytime-feasible (or primal feasible) and (ii) with the possibility to address nonlinear factors on the exchanged data over the network, due to, e.g., utilization of quantized values for more efficient usage of network resources or limited available bandwidth that may cause clipping. In such nonlinear setups, the existing linear methods may fail the feasibility constraint or converge to a sub-optimal solution (or even diverge). Our nonlinear model converges (a) exactly under logarithmic quantization (as a sector-bounded nonlinearity) and (b) with ε-accuracy under uniform quantization (as a non-sector-bounded nonlinearity). In the uniform quantization case, we find the quantization level to ensure convergence to the ε-neighborhood of the optimizer and meet certain ε-accuracy. Our solution paves the way for the use of bandwidth-limited or fast bandwidth-efficient algorithms subject to quantized values or to address the trade-off between ε-accuracy and the limit on the network bandwidth. We derive the *general* sufficient condition on the nonlinear mapping to preserve all-time feasibility in presence of latency and nonlinearities, and converge to the exact optimizer or within its ε-neighborhood. Further, (iii) we take possible data-exchange delays into account and provide two solutions for known and unknown heterogeneous time delays over the network. We explicitly find a (sufficient) max bound on the time-varying delays to *not violate* the algorithm convergence (for a given step rate). Our delay-tolerant algorithm leads to *no feasibility gap* under a general heterogeneous framework (and odd sign-preserving nonlinearities). Further, (iv) this work accounts for possible change and dis-connectivity of the network, i.e., *uniform-connectivity* instead of *all-time* network connectivity as in [1], [18], [24]. We provide (quadratic) CPU scheduling subject to quantized data transmission as an example application, even though the solution works for general non-quadratic models. To our best knowledge, no work in the literature addresses the contributions (i)-(iv) altogether.

C. PAPER ORGANIZATION

Section II states the problem, definitions, and preliminary lemmas. Sections III and IV provide the proposed distributed

nonlinear protocols and the proof of feasibility and convergence under latency. Section V presents the simulation results and Section VI concludes the paper.

II. PROBLEM STATEMENT

General Notation: $\|\cdot\|_2$ denotes the 2-norm. ";" denotes the column vector concatenation. The gradient is defined as $\nabla F(\mathbf{x}) := [\frac{df_1(x_1)}{dx_1}; \ldots; \frac{df_n(x_n)}{dx_n}]$. $\prec, \preceq, \succ, \succeq$ denote the element-wise version of $<, \leq, >, \geq$ operator for vectors. span$\{\cdot\}$ denotes the linear span of a vector. $\mathbf{1}_n$ and $\mathbf{0}_n$ are vectors of all 1 s and 0 s of size n, respectively. RHS and LHS abbreviate right-hand-side and left-hand-side (of an equation). $(\cdot)^\top$ denotes the transpose.

The constrained optimization problem considered in this paper is in the following general mathematical form:

$$\mathcal{P}_0: \quad \min_{\mathbf{y}} \ \widehat{F}(\mathbf{y}) = \sum_{i=1}^{n} \widehat{f}_i(y_i), \ \text{s.t.} \sum_{i=1}^{n} a_i y_i = b, \qquad (1)$$

with $\mathbf{y} = [y_1; \ldots; y_n] \in \mathbb{R}^n$ and y_i as the resource assigned (or to be assigned) to the agent i. The fixed parameter $b \in \mathbb{R}$ represents the fixed amount of total resources, and $\mathbf{a} = [a_1; \ldots; a_n] \in \mathbb{R}^n_+$ is a general weighting vector. The function $\widehat{f}_i(y_i)$ at agent i represents the cost as a function of assigned resources to agent i. By change of variable as $x_i = a_i y_i$, the problem turns into the following simpler (sum-preserving) form:

$$\mathcal{P}_1: \quad \min_{\mathbf{x}} \ F(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i), \ \text{s.t.} \sum_{i=1}^{n} x_i = b, \qquad (2)$$

where $f_i(x_i) = \widehat{f}_i(\frac{x_i}{a_i})$. The cost functions $f_i : \mathbb{R} \mapsto \mathbb{R}$ are strongly-convex[1] and smooth at all agents. This is defined later in the following assumption. We make the following assumption on the objective function.

Assumption 1: The local objectives $f_i(x_i) : \mathbb{R} \to \mathbb{R}$, $i \in \{1, \ldots, n\}$ are strongly convex, proper, and closed with locally Lipschitz derivatives such that $2v < \frac{d^2 f_i(x_i)}{dx_i^2} < 2u$ with $u \geq v > 0$.

Note, however, that the assumption of strong convexity is for determining the algorithm's convergence rate, while strict convexity $0 < \frac{d^2 f_i(x_i)}{dx_i^2}$ is sufficient for the proof of convergence. For quadratic cost strong and strict convexity are equivalent. The ratio $\frac{u}{v} \geq 1$ in Assumption 1 is referred to as the *condition number* of f_i and, for example, equals to 1 for quadratic objective functions (e.g., for CPU scheduling), which may affect the rate of convergence in general distributed optimization problems [34].

In some applications, there are certain box constraints on the states as $\underline{m}_i \leq x_i \leq \overline{M}_i$. One can remove such constraints in $\mathcal{P}_1$ by adding some convex penalty functions [35] or barrier functions [18], [36] (as discussed later). Recall that the sum of the strongly convex $f_i(\cdot)$ and a convex penalty function is strongly convex[2]. In general, the penalty and barrier functions are convex but not necessarily quadratic, and adding them to the objective function makes it non-quadratic. Therefore, such problems cannot be addressed by general consensus-based solutions assuming a quadratic cost model, e.g., the solution by [1]. Some examples of general non-quadratic costs are given in [38] for linear dynamics, where no node/link nonlinear constraint is addressed.

In distributed resource allocation, the idea is to assign resources to the agents in order to solve $\mathcal{P}_1$ in a distributed fashion and based on the local data exchange in the neighborhood of agents[3] (see examples in Section V). However, in practical applications, some constraints on the states or nonlinearities on the agents' dynamics may affect the convergence and solution optimality; for example, when the states take discrete (quantized) values or the communication bandwidth is limited. The main contribution of this work is to address how such possible nonlinearities and constraints can be addressed in the proposed distributed solution. Further, the conditions to reach the exact optimizer of $\mathcal{P}_1$ need to be defined. For example, suppose the exact optimizer cannot be reached under certain conditions. In that case, we need to determine the ε-bound on the convergence, i.e., to define the furthest distance to the optimizer $\mathbf{x}^*$ that the algorithm may converge to (known as the ε-neighborhood bound [39]).

In many existing solutions, participating nodes are assumed to be interconnected with undirected communication links. This means that the network topology forms a connected undirected graph. Note, however, that the results of this paper are suitable for balanced dynamic directed graphs as well, where the network topology may change over time, i.e., our results are valid over uniformly-connected digraphs with balanced (not necessarily bi-stochastic) weights on the incoming and outgoing links. Considering (possibly delayed) information exchange due to time delays while simultaneously handling anytime-feasibility is another contribution of our work. Anytime (or all-time) feasibility implies that the coupling resource-demand constraint in (2) holds at all times and at any termination point of our algorithm.

A. PRELIMINARY DEFINITIONS AND LEMMAS

The network of agents is represented by graph $\mathcal{G}$ with weight matrix W. Define its Laplacian matrix as $L = D - W$ with $D = \text{diag}[\sum_{j=1}^{n} W_{ij}]$ and positive link weights $W_{ij} > 0$.

Assumption 2: Network $\mathcal{G}(k)$ is weight-balanced, i.e., $\mathbf{1}_n^\top W(k) = (W(k)\mathbf{1}_n)^\top$. Further, $W_{ii} = 0$ and $W_{ij} > 0$. The union of the network over every finite time-interval of length

[1] Our results are valid for general *strictly-convex* smooth functions. For the proof of convergence strict convexity is sufficient, while the assumption on strong-convexity is for the purpose of determining the convergence *rate*.

[2] In the presence of general local constraints, the transformation from (1) to (2) by change of variables needs certain condition regarding the composition convexity to be addressed. See [37, Section 3.2.4] for details.

[3] We assume synchronous clocks (and communication) over the network as in many other existing literature.

B, i.e., $\mathcal{G}_B = \bigcup_k^{k+B} \mathcal{G}(k)$ for $k \geq 0$, is strongly-connected, which is known as *uniform-connectivity* or *B-connectivity*[4].

Such weight-balanced digraphs (and their weight matrices W) can be designed using, e.g., the algorithm in [40].

Lemma 1: For a weight-balanced graph $\mathcal{G}$, its Laplacian matrix L is positive semi-definite. Let $\mathbf{y}_1 \in \mathbb{R}^n$, $\overline{\mathbf{y}}_1 := \mathbf{y}_1 - \frac{\mathbf{1}_n^\top \mathbf{y}_1}{n}\mathbf{1}_n$, and λ_n, λ_2 as the largest and smallest non-zero (real) eigenvalue of $L_s = \frac{L+L^\top}{2}$. Then, the following *Laplacian disagreement* function satisfies,

$$\mathbf{y}_1^\top L \mathbf{y}_1 = \mathbf{y}_1^\top L_s \mathbf{y}_1 = \overline{\mathbf{y}}_1^\top L_s \overline{\mathbf{y}}_1, \tag{3}$$

$$\lambda_2 \|\overline{\mathbf{y}}_1\|_2^2 \leq \mathbf{y}_1^\top L_s \mathbf{y}_1 \leq \lambda_n \|\overline{\mathbf{y}}_1\|_2^2 \tag{4}$$

Further, given $\mathbf{y}_2 = g(\mathbf{y}_1) \in \mathbb{R}^n$ as a (element-wise) *monotonic* function of $\mathbf{y}_1$ such that, for ith element of $\mathbf{y}_1, \mathbf{y}_2$, $0 < \kappa \leq \frac{\mathbf{y}_{2i}}{\mathbf{y}_{1i}} \leq \mathcal{K}$ and $(\mathbf{y}_{2i} - \mathbf{y}_{2j})(\mathbf{y}_{1i} - \mathbf{y}_{1j}) \geq 0$ for all i, j, we have

$$\lambda_2 \kappa \|\overline{\mathbf{y}}_1\|_2^2 \leq \mathbf{y}_2^\top L_s \mathbf{y}_1 \leq \lambda_n \mathcal{K} \|\overline{\mathbf{y}}_1\|_2^2 \tag{5}$$

Proof: The proof of (3) and (4) follows from [41]. We prove (5) in the following.

$$\mathbf{y}_2^\top L_s \mathbf{y}_1 = \overline{\mathbf{y}_2}^\top L_s \overline{\mathbf{y}_1} = \frac{1}{2}\sum_{i,j=1}^{n} \overline{W}_{ij}(\mathbf{y}_{2i} - \mathbf{y}_{2j})(\mathbf{y}_{1i} - \mathbf{y}_{1j}) \tag{6}$$

with symmetric matrix $\overline{W}$ defined as $\frac{W+W^\top}{2}$. The first equality above follows from the fact that $\mathbf{1}_n^\top L_s = L_s \mathbf{1}_n = \mathbf{0}_n$. Following the (element-wise) monotonic property of $\mathbf{y}$ with respect to $\mathbf{x}$,

$$\begin{aligned}\kappa(\mathbf{y}_{1i} - \mathbf{y}_{1j})(\mathbf{y}_{1i} - \mathbf{y}_{1j}) &\leq (\mathbf{y}_{2i} - \mathbf{y}_{2j})(\mathbf{y}_{1i} - \mathbf{y}_{1j}) \\ &\leq \mathcal{K}(\mathbf{y}_{1i} - \mathbf{y}_{1j})(\mathbf{y}_{1i} - \mathbf{y}_{1j})\end{aligned}$$

Using the above in (6) along with (4) proves (5). ■

For more information on the above, the notion of mirror digraphs in consensus literature is insightful.

Corollary 1: For uniformly-connected network $\mathcal{G}_B$ with $B > 0$ satisfying Assumption 2, Lemma 1 can be restated for its Laplacian L_B and its largest/smallest non-zero eigenvalue $\lambda_{nB}, \lambda_{2B}$. In (3)–(5) holds for any symmetric positive semi-definite matrix, for example $\mathcal{L} = L^\top L$, satisfying $\mathcal{L}\mathbf{1}_n = \mathbf{0}_n$ and $\mathbf{1}_n^\top \mathcal{L} = \mathbf{0}_n^\top$. Then, $\mathbf{x}^\top \mathcal{L} \mathbf{x} \leq \lambda_n^2 \|\overline{\mathbf{x}}\|_2^2$.

Definition 1: Define $\mathcal{S}_\mathbf{b} = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{1}_n^\top \mathbf{x} = \mathbf{b}\}$ as the set of all feasible values for $\mathbf{x}$.

Lemma 2: Under Assumption 1, $\mathcal{P}_1$ has a *unique* feasible optimizer $\mathbf{x}^* \in \mathcal{S}_\mathbf{b}$ satisfying $\nabla F(\mathbf{x}^*) = \psi^* \mathbf{1}_n$, $\psi^* \in \mathbb{R}$, and $\nabla F(\mathbf{x}) \notin \text{span}\{\mathbf{1}_n\}$, $\forall \mathbf{x} \neq \mathbf{x}^*, \mathbf{x} \in \mathcal{S}_\mathbf{b}$.

Proof: The proof follows from the strong-convexity of $F(\mathbf{x})$ in Assumption 1 and the KKT condition [42]. For completeness, we give another proof based on level-set analysis. Define the level set for $\gamma \in \mathbb{R}$ as $L_\gamma(F) := \{\mathbf{x} \in \mathbb{R}^n | F(\mathbf{x}) \leq \gamma\}$. Assumption 1 implies that all the level sets of $F(\cdot)$ are strongly convex [42], and thus, only one, say $L_\gamma(F)$, touches the affine feasibility constraint facet $\mathcal{S}_\mathbf{b}$ at only one point, say $\mathbf{y}$. Then, $\nabla F(\mathbf{y})$ is orthogonal to the facet $\mathcal{S}_\mathbf{b}$, and $\nabla F(\mathbf{y}) \in \text{span}\{\mathbf{1}_n\}$. For two points $\mathbf{z}, \mathbf{y} \in \mathcal{S}_\mathbf{b}$ on level sets $\gamma_1 = F(\mathbf{z}) > F(\mathbf{y}) = \gamma_2$, from [27, Lemma 1],

$$(\nabla F(\mathbf{z}) - \nabla F(\mathbf{y}))^\top (\mathbf{y} - \mathbf{z}) > 0. \tag{7}$$

By *contradiction* consider both $\nabla F(\mathbf{y}) \in \text{span}\{\mathbf{1}_n\}$ and $\nabla F(\mathbf{z}) \in \text{span}\{\mathbf{1}_n\}$. This implies either two points (i) on the same level set $L_\gamma(F)$, $\gamma = F(\mathbf{y}) = F(\mathbf{z})$ both adjacent to the affine feasibility constraint set $\mathcal{S}_\mathbf{b}$, or (ii) on two level sets $L_{\gamma_1}(F), L_{\gamma_2}(F)$ with $\gamma_1 = F(\mathbf{y}) \neq F(\mathbf{z}) = \gamma_2$ adjacent to $\mathcal{S}_\mathbf{b}$. Since $\mathcal{S}_\mathbf{b}$ is linear, (i) contradicts the Assumption 1 on the strong convexity of the level sets. In case (ii), $\mathbf{y}, \mathbf{z} \in \mathcal{S}_\mathbf{b}$ implies $\mathbf{1}_n^\top(\mathbf{y} - \mathbf{z}) = 0$ and $(\nabla F(\mathbf{y}) - \nabla F(\mathbf{z}))^\top (\mathbf{y} - \mathbf{z}) = 0$ which contradicts (7). ■

Note that $\mathbf{x}^*$ defined in the above is assumed to follow the box constraints, i.e., $\underline{m}_i \leq x_i^* \leq \overline{M}_i$ for all i.

Lemma 3 ([42], [43]): Consider a *strictly-convex* continuous function $F : \mathbb{R}^n \to \mathbb{R}$, two points $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$, and $\delta\mathbf{x} := \mathbf{x}_1 - \mathbf{x}_2$. There exists $\widehat{\mathbf{x}} := \alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2$, $0 < \alpha < 1$ such that,

$$F(\mathbf{x}_1) = F(\mathbf{x}_2) + \nabla F(\mathbf{x}_2)^\top \delta\mathbf{x} + \frac{1}{2}\delta\mathbf{x}^\top \nabla^2 F(\widehat{\mathbf{x}})\delta\mathbf{x}.$$

Then, from Assumption 1, for *strongly convex* function F,

$$F(\mathbf{x}_1) \geq F(\mathbf{x}_2) + \nabla F(\mathbf{x}_2)^\top \delta\mathbf{x} + v\delta\mathbf{x}^\top \delta\mathbf{x} \tag{8}$$

$$F(\mathbf{x}_1) \leq F(\mathbf{x}_2) + \nabla F(\mathbf{x}_2)^\top \delta\mathbf{x} + u\delta\mathbf{x}^\top \delta\mathbf{x} \tag{9}$$

In (8)–(9) are also known as quadratic lower and upper bound equations. For Lipschitz continuous functions (9) refers to the generalized Cauchy–Schwarz inequality.

Lemma 4: Define $\xi(\mathbf{x}) := \nabla F(\mathbf{x}) - \frac{1}{n}\sum_{i=1}^n \frac{df_i(x_i)}{dx_i}\mathbf{1}_n$. Then, for any $\mathbf{x} \in \mathcal{S}_b$,

$$\frac{1}{4u}\xi^\top \xi \leq \overline{F} \leq \frac{1}{4v}\xi^\top \xi, \tag{10}$$

$$\nabla F(\mathbf{x})^\top \delta\mathbf{x} = \xi(\mathbf{x})^\top \delta\mathbf{x}. \tag{11}$$

with $\overline{F} := F(\mathbf{x}) - F^*$ and F^* as the optimal cost $F(\mathbf{x}^*)$.

Proof: The proof directly follows from Lemma 2. See, e.g., [10] for the proof of (10). The proof of (11) is as follows,

$$\nabla F(\mathbf{x})^\top \delta\mathbf{x} = \xi^\top \delta\mathbf{x} + \frac{1}{n}\sum_{i=1}^n \frac{df_i(x_i)}{dx_i}\mathbf{1}_n^\top \delta\mathbf{x} = \xi^\top \delta\mathbf{x}$$

where the latter follows from $\mathbf{1}_n^\top \delta\mathbf{x} = \mathbf{1}_n^\top \mathbf{x}_1 - \mathbf{1}_n^\top \mathbf{x}_2$ for any two feasible $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{S}_b$. Putting $\mathbf{x}_2 = \mathbf{x}^*$ and considering $\mathbf{x}_1$ as any feasible $\mathbf{x} \in \mathcal{S}_b$ (11) follows. ■

Note that for notation simplicity, we dropped the dependence on $\mathbf{x}$ (also in the rest of the paper unless needed). The next lemma follows from Assumption 1 and the fact that strong-convexity implies strict convexity.

[4]In this work, the network topology is assumed time-varying, in general. For notation simplicity, we drop the dependence of $\mathcal{G}$, W, and other network parameters on time-index k unless it is required for clarification purposes.

Lemma 5: Let Assumption 1 hold. For any $\mathbf{x} \in \mathcal{S}_b$ define residual $\overline{F} = F(\mathbf{x}) - F^*$ and $\xi(\mathbf{x})$ as in Lemma 4. Then,

$$v\|\mathbf{x} - \mathbf{x}^*\|_2^2 \leq \overline{F} \leq u\|\mathbf{x} - \mathbf{x}^*\|_2^2, \tag{12}$$

$$\frac{\|\xi\|_2}{2u} \leq \|\mathbf{x} - \mathbf{x}^*\|_2 \leq \frac{\|\xi\|_2}{2v}. \tag{13}$$

Proof: From Lemma 3, substituting $\mathbf{x}_1 = \mathbf{x}$ and $\mathbf{x}_2 = \mathbf{x}^*$ we get,

$$\nabla F(\mathbf{x}^*)^\top \delta\mathbf{x} + v\delta\mathbf{x}^\top\delta\mathbf{x} \leq \overline{F} \leq \nabla F(\mathbf{x}^*)^\top\delta\mathbf{x} + u\delta\mathbf{x}^\top\delta\mathbf{x}$$

From Lemma 4, $\nabla F(\mathbf{x}^*)^\top\delta\mathbf{x} = \xi(\mathbf{x}^*)^\top\delta\mathbf{x} = 0$ since $\xi(\mathbf{x}^*) = \mathbf{0}_n$ from the definition. This gives (12) which, along with (10) and taking the square roots results in (13). ■

III. PROPOSED DISCRETE-TIME NONLINEAR SOLUTION

In this section, we introduce a 1st-order protocol to update the state of agents at every time-step k, while considering possible nonlinear models on the data transmissions. We consider a group of n agents sharing information over nonlinear (possibly) delayed links. Following a common assumption in the literature, we assume synchronized clocks over the multi-agent network. This can be implemented by e.g., the fully-distributed algorithms proposed in [44], [45], [46] for synchronization over (wireless) sensor network. At time-step k, every agent i shares $\frac{df_i(x_i)}{dx_i}$ with its out-neighbors $j \in \mathcal{N}_i^+$, and agent j receives $\varphi_i := g(\frac{df_i(x_i)}{dx_i})$ from in-neighbors $i \in \mathcal{N}_j^-$, where $g(\cdot)$ represents a nonlinear mapping due to, e.g., signal *clipping* or *logarithmic quantization* over the channel. The (delay-free) information update at agent i is,

$$x_i(k+1) = x_i(k) - \eta \sum_{j\in\mathcal{N}_i^-} W_{ij}(\varphi_i(k) - \varphi_j(k)) \tag{14}$$

with $k \in \mathbb{Z}^{\geq 0}$ as the time-index, $\eta > 0$ as the step size, and $W = [W_{ij}]$ satisfying Assumption 2. In terms of implementation, at the beginning of each time slot, each node i receives the states of its in-neighbors $j \in \mathcal{N}_i^-$ and multi-casts (or broadcasts) its own state to its out-neighbors $j \in \mathcal{N}_i^+$. Then, it updates its state $x_i(k+1)$ based on the received information (and its own previous state $x_i(k)$) as in (14). Similar to the existing literature, we assume collision-free packets and contention mechanisms to resolve this issue over the networked busses, where the details are out of the scope of this work and skipped here.

The vector form of the coordination protocol (14) is

$$\mathbf{x}(k+1) = \mathbf{x}(k) - \eta L\varphi(k), \tag{15}$$

with L as the graph Laplacian. One can consider a more general formulation (over undirected weight-symmetric networks) by adding a post-processing step as a node nonlinearity along with the nonlinearity on the links in (14). This gives a more general formulation as,

$$x_i(k+1) = x_i(k) - \eta \sum_{j\in\mathcal{N}_i^-} W_{ij}g(\varphi_i(k) - \varphi_j(k)), \tag{16}$$

where the nonlinearities $g(\cdot)$ in (14) and (16) are different in general, however, both satisfy the assumption below.

Assumption 3: The nonlinear mapping $g : \mathbb{R} \to \mathbb{R}$ is sign-preserving and odd, i.e., $g(z)z > 0$ for $z \neq 0$, $g(0) = 0$. Further,

$$\kappa \leq \frac{g(z)}{z} \leq \mathcal{K}, \tag{17}$$

and $\lim_{z\to 0}\frac{g(z)}{z} \neq 0$ implying that $g(\cdot)$ is "strongly" sign-preserving. Further $g(\cdot)$ is monotonically non-decreasing.

Example nonlinearities satisfying Assumption 3 are logarithmic quantization [33], [47] and saturation (or clipping) [27]. In this work we more focus on quantization which is an inherent property of the network and at all the links. It is typically assumed that these nonlinearities are generally the same at all links; see some more examples for nonlinear robust consensus in [31]. As we see later in this section, the convergence of our algorithm under dynamics (14) and (16) is proved under fixed step size η. This is a privilege in terms of convergence rate over diminishing step sizes in some algorithms as in [20]; see some detailed discussions on this in [30]. Note that the exact convergence rate of (16) (and (14)) depends on the choice of the nonlinearity $g(\cdot)$. For example, it is even possible to achieve convergence in fixed or finite time by considering $g(\cdot)$ as sign-based nonlinearities; see, for example, [4], [27], [48].

One immediate implication of Assumption 3 is,

$$\kappa^2\nabla F^\top\nabla F \leq \varphi^\top\varphi \leq \mathcal{K}^2\nabla F^\top\nabla F, \tag{18}$$

$$\kappa\nabla F^\top\nabla F \leq \nabla F^\top\varphi \leq \mathcal{K}\nabla F^\top\nabla F, \tag{19}$$

where we drop the dependence on time-index k for notation simplicity (also in the rest of the paper unless where needed).

A. PROOF OF FEASIBILITY AND CONVERGENCE

First we discuss anytime-feasibility, i.e., under initialization $\mathbf{x}(0) \in \mathcal{S}_\mathbf{b}$ the solution preserves its feasibility at every k (referred to as sum-preserving property). Under similar box constraints at all nodes, a simple local initialization is $\frac{b}{n}$. Under heterogeneous box constraints, one can use existing results for establishing a feasible initialization in a distributed way. For example, [10, Algorithm 2], provides a finite-time algorithm to re-adjust the initial guesses (within the box constraints) even for a network of time-varying sizes.

Lemma 6 (Feasibility & Uniqueness): Let Assumptions 1, 2, and 3 hold. By any feasible initialization $\mathbf{x}(0) \in \mathcal{S}_\mathbf{b}$, the solution under dynamics (14) remains feasible, i.e., $\mathbf{x}(k) \in \mathcal{S}_\mathbf{b}, \forall k > 0$. Further, $\nabla F(\mathbf{x}^*) = \psi^*\mathbf{1}_n$ with $\mathbf{x}^*$ as equilibrium under dynamics (14) and $\psi^* \in \mathbb{R}$ from Lemma 2.

Proof: $\mathbf{x}(0) \in \mathcal{S}_\mathbf{b}$ implies that $\sum_{i=1}^n x_i(0) = b$. Following Assumptions 2 and 3, we have $\mathbf{1}_n^\top L\varphi(k) = \mathbf{0}_n$ where the gradient is well-defined from Assumption 1. Note that this holds irrespective of network connectivity and is a direct result of symmetric weights and oddness of $g(\cdot)$. Then, from (15), $\sum_{i=1}^n x_i(k+1) = \sum_{i=1}^n x_i(k) = b$ for all $k \geq 0$. Next, assume $\nabla F(\mathbf{x}^*) = \boldsymbol{\theta} \notin \text{span}\{\mathbf{1}_n\}$ and, thus, there exist nodes

α and $\beta \in \mathcal{N}_\alpha^-$ over $\mathcal{G}_B(k)$ such that $\boldsymbol{\theta}_\alpha > \boldsymbol{\theta}_\beta$. From (14), $x_\alpha(k+1) < x_\alpha(k)$, implying that such $\mathbf{x}^*$ is not an equilibrium of (14), which is a contradiction. Thus, for the equilibrium $\mathbf{x}^*$ we have $\nabla F(\mathbf{x}^*) \in \text{span}\{\mathbf{1}_n\}$. Using Lemma 2 and anytime-feasibility in Lemma 6, $\forall \mathbf{x} \in \mathcal{S}_b$ there is no other $\mathbf{x} \neq \mathbf{x}^*$ satisfying $\nabla F(\mathbf{x}) \in \text{span}\{\mathbf{1}_n\}$. This implies that $\mathbf{x}^*$ with $\nabla F(\mathbf{x}^*) = \psi^* \mathbf{1}_n$ is the unique equilibrium of (14). ■

Theorem 1 (Convergence): Initializing from $\mathbf{x}(0) \in \mathcal{S}_\mathbf{b}$ and under Assumptions 1–3, dynamics (14) converges to $\mathbf{x}^*$ with $\nabla F(\mathbf{x}^*) =: \psi^* \mathbf{1}_n$ (as the optimal solution of $\mathcal{P}_1$) for small enough step-rate η (see the bound in Theorem 2 and 3).

Proof: Consider positive Lyapunov-type function $\overline{F}(k) := F(\mathbf{x}(k)) - F^*$ (as in Lemma 4 and 5) representing the *residual* cost. We prove $\overline{F}(k+1) < \overline{F}(k)$ under dynamics (14) for $\mathbf{x}(k) \neq \mathbf{x}^*$ and $\overline{F}(\mathbf{x}^*) = 0$. For two feasible states $\mathbf{x}(k+1), \mathbf{x}(k) \in \mathcal{S}_b$ define $\delta\mathbf{x}(k) := \mathbf{x}(k+1) - \mathbf{x}(k)$. From Lemma 3 we need to show that

$$\nabla F^\top \delta\mathbf{x} + u\delta\mathbf{x}^\top \delta\mathbf{x} \leq 0, \tag{20}$$

$$-\eta \nabla F^\top L\varphi + u\eta^2 \varphi^\top L^\top L\varphi \leq 0, \tag{21}$$

where the latter follows from dynamics (15). From Assumption 3 and (18)–(19), and Lemma 1 (Corollary 1 for uniformly connected networks), the above is satisfied if

$$(-\kappa\eta\lambda_2 + u\lambda_n^2 \mathcal{K}^2 \eta^2)\xi^\top \xi \leq 0, \tag{22}$$

where the strict inequality above holds for

$$\eta < \frac{\kappa\lambda_2}{u\lambda_n^2 \mathcal{K}^2} =: \overline{\eta} \tag{23}$$

and for $\xi = \mathbf{0}$ holds the equality. In other words, $\overline{F}(\mathbf{x}(k+1)) \leq \overline{F}(\mathbf{x}(k))$ and from Lemma 2 the strict equality uniquely holds for

$$\overline{F}(\mathbf{x}(k+1)) = \overline{F}(\mathbf{x}(k)) \iff \frac{df_i}{dx_i} = \frac{df_j}{dx_j} = \psi^*, \ \forall i, j. \tag{24}$$

Therefore, under the proposed dynamics (14), the function $\overline{F}$ is decreasing in time[5] till $\mathbf{x}$ reaches the unique equilibrium point $\mathbf{x}^*$ satisfying (24), which is the optimizer of $\mathcal{P}_1$ and the theorem follows. ■

Note that for non-Lipschitz mapping $g(\cdot)$, a similar line of reasoning results in

$$\eta < \frac{\kappa\lambda_2}{u\lambda_n^2} \frac{\xi^\top \xi}{\overline{\varphi}^\top \overline{\varphi}} =: \overline{\eta} \tag{25}$$

instead of (23) with $\overline{\varphi} = \varphi - \frac{1}{n}\sum_{i=1}^n \varphi_i \mathbf{1}_n$.

Lemma 7: Let Assumptions 1, 2, and 3 hold and $\mathbf{x}(0) \in \mathcal{S}_b$. Following the notations in Theorem 1, for $\eta < \overline{\eta}$ the rate of decrease in $\overline{F}(k)$ under protocol (14) is

$$\frac{\overline{F}(k+1)}{\overline{F}(k)} \leq 1 - 4v(\kappa\eta\lambda_2 - u\lambda_n^2 \mathcal{K}^2 \eta^2) \tag{26}$$

[5]Recall that to prove $\overline{F}(\mathbf{x}(k+1)) < \overline{F}(\mathbf{x}(k))$ for $\mathbf{x} \neq \mathbf{x}^*$, strict convexity of f_i is sufficient. The strong-convexity assumption is for determining the bound on the linear rate of convergence.

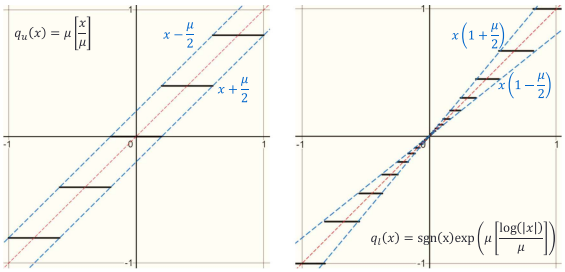

FIGURE 1. Two quantization approaches: (left) uniform, and (right) logarithmic quantization with level μ. Logarithmic quantizer $q_l(\cdot)$ is "strongly" sign-preserving as $\lim_{z\to 0} \frac{q_l(z)}{z} \geq (1 - \frac{\mu}{2}) > 0$, in contrast to uniform quantizer with $\frac{q_u(z)}{z} = 0$ for $-\frac{\mu}{2} \leq z < \frac{\mu}{2}$.

and $\mathbf{x}(k)$ converges to $\mathbf{x}^*$ with the rate

$$\frac{\|\mathbf{x}(k) - \mathbf{x}^*\|_2^2}{\|\mathbf{x}(0) - \mathbf{x}^*\|_2^2} \leq \frac{u}{v}(1 - 4v\kappa\eta\lambda_2 + 4vu\lambda_n^2 \mathcal{K}^2 \eta^2)^k. \tag{27}$$

Proof: From Lemma 4 and 5,

$$\delta\overline{F} = \overline{F}(k+1) - \overline{F}(k) \leq \nabla F^\top \delta\mathbf{x} + u\delta\mathbf{x}^\top \delta\mathbf{x}$$

at time k. Following (22) for $\eta < \overline{\eta}$,

$$\delta\overline{F} \leq (-\kappa\eta\lambda_2 + u\lambda_n^2 \mathcal{K}^2 \eta^2)\xi^\top \xi \leq 0$$

and using the RHS of (10) we have $4v\overline{F} \leq \xi^\top \xi$ which results in (26). Here, we used the fact that the term $-\kappa\eta\lambda_2 + u\lambda_n^2 \mathcal{K}^2 \eta^2$ is non-positive. Combining (26) with (12) gives,

$$\begin{aligned}\|\mathbf{x}(k) - \mathbf{x}^*\|_2^2 &\leq \frac{\overline{F}(k)}{v} \\ &\leq \frac{1}{v}(1 - 4v\kappa\eta\lambda_2 + 4vu\lambda_n^2 \mathcal{K}^2 \eta^2)^k \overline{F}(0) \\ &\leq \frac{u}{v}(1 - 4v\kappa\eta\lambda_2 + 4vu\lambda_n^2 \mathcal{K}^2 \eta^2)^k \|\mathbf{x}(0) - \mathbf{x}^*\|_2^2\end{aligned}$$

For a B-connected network, from Corollary 1, the proof can be stated over time scale of size B, i.e., one can prove $\overline{F}(k+B) < \overline{F}(k)$ for the B-connected network $\mathcal{G}_B$ with eigenvalues $\lambda_{2B}, \lambda_{nB}$. Then, the proof similarly follows. ■

Note that the RHS of (26) and (27) give a bound on the convergence rate as a function of η. The RHS of (27) is quadratic and, for example, gives the tightest bound for $\eta = \frac{\overline{\eta}}{2} = \frac{\kappa\lambda_2}{2u\lambda_n^2 \mathcal{K}^2}$. Similar results are given for unconstrained distributed optimization, e.g., in [30].

B. UNIFORM QUANTIZATION AND ε-ACCURACY

Next, we consider the case that nonlinear mapping $g(\cdot)$ is sign-preserving, but not "strongly" sign-preserving. A simple example is when $\frac{dg}{dz}|_{z=0} = 0$ as in uniform quantization (in case of finite packet size for the networking links), see Fig. 1. Uniform quantization is motivated by recent digital communication devices with a limited number of bit-transmissions. The number of bits (quantization level) defines the efficiency and accuracy and depends on the limitations of the communication equipment. For this case, the formulation (14) can be written

as,

$$x_i(k+1) = x_i(k) - \eta \sum_{j\in\mathcal{N}_i^-} W_{ij}\left(q_u\left(\frac{df_i(k)}{dx_i}\right) - q_u\left(\frac{df_j(k)}{dx_j}\right)\right), \tag{28}$$

with $W_{ij} > 0$ for $j \in \mathcal{N}_i^-$, in general. One can consider (possible) non-negative integer weights $W_{ij} \in \mathbb{N}$, e.g., using the distributed integer weight-balancing algorithm in [40]. These integer weights result in *quantized* (or discrete) allocated values x_i (with level μ), e.g., for task allocation in CPU scheduling [1].

Remark 1: For $g(\cdot)$ representing the *uniform quantizer*, we have $\frac{dg(z)}{dz}|_{z=0} = 0$ and $g(z) = 0$ for $-0.5\mu < z < 0.5\mu$. This implies that for $-0.5\mu\mathbf{1}_n \prec \nabla F - \psi^*\mathbf{1}_n \prec 0.5\mu\mathbf{1}_n$ we have $\mathbf{x}(k+1) = \mathbf{x}(k)$ in (28). In this case, the solution reaches the ε-neighborhood of $\mathbf{x}^*$ instead. Then, one can define this ε as follows. Since $-0.5\mu + \psi^* < \frac{df_i}{dx_i} < \psi^* + 0.5\mu$ and following the definition of ξ, we have $|\xi| \prec 0.5\mu\mathbf{1}_n$ and,

$$\xi^\top\xi < 0.25\mu^2\mathbf{1}_n^\top\mathbf{1}_n = 0.25n\mu^2$$

From (13),

$$\|\mathbf{x}(k) - \mathbf{x}^*\|_2 \le \frac{\|\xi\|_2}{2v} < \frac{\sqrt{n}\mu}{4v} =: \bar{\varepsilon} \tag{29}$$

which gives an estimate that how close we can get to the optimizer $\mathbf{x}^*$.

The above remark implies that by choosing fine uniform quantization (i.e., smaller μ) the solution can get arbitrarily close to the optimizer $\mathbf{x}^*$. In this perspective, one can define the notion of ε-accuracy [39], i.e., the upper-bound on the quantization level μ that guarantees convergence to the ε-neighborhood of the optimizer $\mathbf{x}^*$. The ε-accuracy of the solution, then, follows (29), i.e., for the quantization level μ the solution is guaranteed to reach the ε-neighborhood of $\mathbf{x}^*$ for $\varepsilon \le \frac{\sqrt{n}\mu}{4v}$.

C. DISCUSSIONS AND MORE APPLICATIONS

There exist many other applications for finite-sum resource allocation; many works [10], [49], [50] are dedicated to generator coordination over smart grids (known as the *economic dispatch problem*). In contrast to many existing solutions which are limited to the *quadratic* cost model, as in economic dispatch, CPU scheduling, and general consensus problems [51], the cost function in this paper only needs to be strongly-convex. This allows to consider the solution for many applications with non-quadratic costs, e.g., the cost function in [38], or to add extra objective terms to address, e.g., penalty terms for the so-called box-constraints $\sigma[\mathbf{x}_i - \overline{M}_i]^+ + \sigma[\underline{m}_i - \mathbf{x}_i]^+$, with $[u]^+ = \max\{u, \mathbf{0}\}^c$, $c \in \mathbb{Z}^+$ [35]. Note that $[u]^+$ is smooth for $c \ge 2$, and one can also use smooth equivalents for non-smooth case $c = 1$. In this case, non-quadratic (but strongly-convex) penalty terms in the logarithmic form $\frac{\sigma}{\rho}\log(1 + \exp(\rho u))$ are typically proposed [52] to replace the non-smooth penalties for case $c = 1$. σ is a weighting factor to tune the weight on the box constraint. It can be shown that by setting ρ large enough $L(z, \rho)$ becomes arbitrarily close to $\max\{u, 0\}$; in fact, the maximum gap between the two functions inversely scales with ρ, i.e., $L(z, \rho) - \max\{z, 0\} \le \frac{1}{\rho}$, and the two can become arbitrarily close by selecting ρ sufficiently large [52]. Similarly, some smooth and convex barrier functions are proposed in the literature (e.g., see [18], [36]). Let $h_i(\mathbf{x}_i)$ represent a general local constraint at agent i. Then, example barrier functions are defined as,

$$\mathcal{B}_i(\mathbf{x}_i) := -\log(-h_i(\mathbf{x}_i)), \quad \mathcal{B}_i(\mathbf{x}_i) := \frac{-1}{h_i(\mathbf{x}_i)} \tag{30}$$

which are known as, respectively, the logarithmic and inverse barrier functions. In general the barrier functions satisfy the following: if $h_i(\mathbf{x}_i) \to 0^-$, then $\mathcal{B}_i(\mathbf{x}_i) \to \infty$.

Our results can, further, help to reach a faster rate of convergence by using sign-based dynamics [27] for the nonlinear node-based solution. In discrete-time, such non-Lipschitz dynamics mandate a sufficiently small step rate to reduce the so-called chattering effect. There is a trade-off (to be properly adjusted) between the steady-state residual around the equilibrium and the step size. In applications, one can reach convergence in finite, fixed, or prescribed time irrespective of the chattering (e.g., in continuous-time [27], [48]).

IV. SOLUTION IN THE PRESENCE OF TIME-DELAYS

A. DOUBLE TIME-SCALE SCHEME: UNKNOWN DELAYS

Our first solution is to update the states at a longer time-scale such that at every link one message is delivered. The following assumption (as in [32]) holds on this subsection for time-delay $\tau_{ij}(k)$ on every link (j, i):

Assumption 4:

- $\tau_{ij}(k) \le \overline{\tau}$ where $1 \le \overline{\tau} < \infty$ represents the upper-bound on the time-delays ($\overline{\tau} = 0$ implies no delay). The finite integer bound $\overline{\tau}$ ensures that data from agent i at time k eventually reaches agent j at most in $k + \overline{\tau}$, and also implies no missing packet.
- $\tau_{ij}(k)$ is, in general, heterogeneous, arbitrary, time-variant, and not necessarily known.

The proposed state-update under Assumption 4 is as follows:

$$x_i(\bar{k}+1) = x_i(\bar{k}) - \eta \sum_{j\in\mathcal{N}_i^-} W_{ij}(\varphi_i(\bar{k}) - \varphi_j(\bar{k})) \tag{31}$$

with $\bar{k} = \lfloor \frac{k}{\overline{\tau}+1} \rfloor$ as the new time-scale. In this method, the states get updated not in every time-step k, but every $\overline{\tau} + 1$ time-steps, representing a longer time-scale $\bar{k}$. In other words, after initializing and sending the first messages at step $\bar{k} = k = 0$, the next *state-update* and *communication* occurs at $k = \overline{\tau} + 1$ (i.e., $\bar{k} = 1$) and every $\overline{\tau} + 1$ steps on k afterwards, while states remain unchanged at $k \ne \bar{k}(\overline{\tau} + 1)$. Following Assumption 4 on time-delays, it is clear that over every link (j, i) one data-packet is received by agent i from in-neighbor j over $\overline{\tau} + 1$ steps of k. The feasibility and convergence under the delayed model are proved next.

Theorem 2: Under Assumptions 1, 2, 3, and 4, with $\mathbf{x}(0) \in \mathcal{S}_b$ (feasible initializing), the states $\mathbf{x}(k)$ (and $\mathbf{x}(\bar{k})$) under protocol (31) remain feasible and converge to the optimal solution of (2) for $0 < \eta < \overline{\eta}$.

Proof: The proof of solution feasibility follows similar to Lemma 6 by considering the state-update over time-scale $\bar{k}$. The uniqueness of the optimizer is similar to considering uniform-connectivity of the network over $B + \overline{\tau}$. The convergence to the optimizer follows a similar analysis as in Theorem 1 over the time-scale $\bar{k}$. For two feasible states $\mathbf{x}(\bar{k}+1), \mathbf{x}(\bar{k}) \in \mathcal{S}_b$ define $\delta\mathbf{x}(\bar{k}) =: \mathbf{x}(\bar{k}+1) - \mathbf{x}(\bar{k})$, $\delta F(\bar{k}) =: F(\mathbf{x}(\bar{k}+1)) - F(\mathbf{x}(\bar{k}))$. Then, following the same line of reasoning as in (20)–(25), one can find the same bound on the convergence step rate as $\eta < \overline{\eta}$. ■

Note that the convergence of this double time-scale scheme is slow, as agents need to wait for a while to receive the delayed information and then update their states. However, if delays are known and symmetric over bidirectional links, states can get updated at the *same* time scale. It is clear that this gives faster convergence, as discussed next.

B. UPDATE AT THE SAME TIME-SCALE: KNOWN DELAYS

Our other solution is to update the state at the same time-scale k. We make the following assumptions for this case.

Assumption 5:

- Delays $\tau_{ij}(k)$ are bounded by $\overline{\tau}$, and are arbitrary, possibly heterogeneous at different links, and time-varying, in general.
- The messages over every link (j, i) are timestamped and every agent i knows the time-step agent j sent the information, i.e., $\tau_{ij}(k)$ is known.
- The network $\mathcal{G}$ is undirected with *symmetric* link-weights and time-delays for both sides of the links are the same, i.e., $W_{ij} = W_{ji}$ and $\tau_{ij}(k) = \tau_{ji}(k)$. The uniform connectivity follows similar to Assumption 2.

The state of every agent i is updated based on all the (possibly delayed) information received from neighbours at time $k+1$ *as they arrive*. Note that since the delays are assumed to be heterogeneous, the received information at time k is, in general, sent over the range $[k - \overline{\tau}, k]$ (the last $\overline{\tau}$ time-steps). Also, assuming time-varying delays, it is possible that at time-step k agent i receives more than one packet (from in-neighbour j). This makes the solution more challenging in terms of satisfying anytime feasibility. Recall that, for anytime feasibility, $\sum_{i=1}^n x_i(k+1) = \sum_{i=1}^n x_i(k)$ needs to be hold at every time k, which is satisfied by synchronous messaging over both directions (i, j) and (j, i) of every link. For the same reason, the weights of all bidirectional links are designed symmetrically. We discuss this more in the feasibility analysis in Lemma 8. The proposed single time-scale protocol in the presence of time delays is as follows.

$$x_i(k+1) = x_i(k) - \eta_\tau \sum_{j \in \mathcal{N}_i^-} \sum_{r=0}^{\overline{\tau}} W_{ij}(\varphi_i(k-r) - \varphi_j(k-r))\mathcal{I}_{k-r,ij}(r), \tag{32}$$

where $\mathcal{I}$ is the indicator function,

$$\mathcal{I}_{k,ij}(r) = \begin{cases} 1, & \text{if } \tau_{ij}(k) = r, \\ 0, & \text{otherwise.} \end{cases} \tag{33}$$

Note that $\mathcal{I}_{k-r,ij}(r) \neq 0$ indicates the message received at time k with time-delay $\tau_{ij} = r$ over the link (j, i) (i.e., sent at time $k - r$). In general, we assume $\mathcal{I}_{k-r,ij}(r) \neq 0$ for at least one pair $(i, j \in \mathcal{N}_i^-)$ at every time k. This assumption simply means that at least one message is delivered over the network at every time k, and is only required to ensure that the cost *monotonically* decreases at every time step under the proposed dynamics. However, without this consideration, the solution still converges over time. The following remark relaxes Assumption 5 by using definition (33).

Remark 2: As a follow-up to Assumption 5, in case of *known* but *asymmetric* time-delays at bidirectional links, say $\tau_{ij} \neq \tau_{ji} < \overline{\tau}$, both agents i, j can process their mutual information based on the known max delay $\overline{\tau}$ (or possibly known $\max\{\tau_{ij}, \tau_{ji}\} \leq \overline{\tau}$) on the shared mutual link, i.e., $\mathcal{I}_{k,ij}(\overline{\tau}) = \mathcal{I}_{k,ji}(\overline{\tau}) = 1$ instead of $\mathcal{I}_{k,ij}(\tau_{ij}) = 1$, $\mathcal{I}_{k,ji}(\tau_{ji}) = 1$, $\tau_{ij} \neq \tau_{ji}$. This implies that both agents apply (process) their shared information at the same time. This can be thought as a combination of the two schemes in subsections A and B.

Lemma 8: The solution under (32) and Assumptions 1, 2, 3, and 5 is anytime feasible with unique equilibrium $\mathbf{x}^*$ as the optimizer of $\mathcal{P}_1$.

Proof: Following Assumption 5, for every pair of links (j, i) and (i, j) we have $W_{ij} = W_{ji}$, $\tau_{ij} = \tau_{ji}$, $\mathcal{I}_{k,ij}(\tau_{ij}) = \mathcal{I}_{k,ji}(\tau_{ji}) = 1$, and $\mathcal{I}_{k,ij}(r) = \mathcal{I}_{k,ji}(r) = 0$ for $r \neq \tau_{ij}, \tau_{ji}$. Therefore,

$$W_{ij}(\varphi_i(k-r) - \varphi_j(k-r))\mathcal{I}_{k-r,ij}(r) = -W_{ji}(\varphi_j(k-r) - \varphi_i(k-r))\mathcal{I}_{k-r,ji}(r)$$

This implies that $\sum_{i=1}^n x_i(k+1) = \sum_{i=1}^n x_i(k)$ and the feasibility follows for all $k \geq 0$. The uniqueness follows similar to Theorem 2 by considering uniform-connectivity over $B + \overline{\tau}$. This completes the proof. ■

Theorem 3: Under Assumptions 1, 2, 3, and 5, with $\mathbf{x}(0) \in \mathcal{S}_b$, solution under protocol (32) converges to the optimizer of $\mathcal{P}_1$ for,

$$0 < \eta_\tau < \frac{\overline{\eta}}{\overline{\tau} + 1}. \tag{34}$$

with $\overline{\eta}$ given in (23).

Proof: The proof follows from Lemma 1. First, consider a homogeneous case where $\tau_{ij} = \overline{\tau}$, i.e., agents' states at any iteration k next updates at $k + \overline{\tau} + 1$ (and every $\overline{\tau} + 1$ steps afterwards). The bound on η, then, follows from Theorem (2) and (23). Next, for general (heterogeneous) delays two cases are possible: Case (i), time-invariant (fixed) delays at all links where every node i receives *only one* (possibly) delayed packet from $j \in \mathcal{N}_i^-$ and $\delta\mathbf{x}$ remains the same as in (20)–(25); this gives the same bound as $\eta < \overline{\eta}$. Case (ii), for general *time-varying* delays (satisfying Assumption 4), node i receives *at most* $\overline{\tau} + 1$ delayed packets from the nodes $j \in \mathcal{N}_i^-$; and, thus, $\overline{\eta}$ needs to be down-scaled by $\overline{\tau} + 1$

to ensure convergence. This is because (15) is scaled by $\delta\mathbf{x} = -(\overline{\tau}+1)\eta_\tau L\varphi(k)$ in the proof of Theorem 1 at step k and, following the same line of reasoning as in (21)–(25), $(\overline{\tau}+1)\eta_\tau < \overline{\eta}$ guarantees $\overline{F}(k) < \overline{F}(k-1)$ for $\mathbf{x}(k-1) \neq \mathbf{x}^*$. This completes the proof. ■

C. CONVERGENCE AND OPTIMALITY DISCUSSIONS

Remark 3: The following remarks are noteworthy:

i) For time-invariant delays, one can further relax the upper-bound in (34). In this case, at every time-step $k \geq \overline{\tau}+1$, agent i receives only one packet from agent $j \in \mathcal{N}_i^-$. Following the same line of reasoning as in the proof of Theorem 1 and 3, the solution (32) converges for $\eta_\tau < \overline{\eta}$.

ii) Convergence under protocol (32) is faster than (31) for the same step rate $\eta = \eta_\tau$, since the time-scale $\overline{k}$ is $\overline{\tau}+1$ times longer than time-scale k. However, for general *time-varying* delays, the solution (32) may not necessarily converge for $\frac{\overline{\eta}}{\overline{\tau}+1} \leq \eta_\tau \leq \overline{\eta}$, while solution (31) converges.

iii) Recall that in contrast to logarithmic quantization, uniform quantization is not "strongly" sign-preserving (see Fig. 1 for $0 < \mu < 1$). In this case, the lower-bound in (17) does not hold for any $\kappa > 0$. Similarly, in case that $\frac{g(z)}{z}$ is not upper-bounded, e.g., $g(z) = z|z|^{\nu-1}$ as in finite-time control/consensus [4], the bound (17) in Assumption 3 does not hold. In such cases, and similar sign-preserving odd mappings, the convergence of *discrete-time* protocol (14) to the exact optimizer cannot be guaranteed and one can prove convergence to an ε-neighborhood of the optimizer, e.g., see [53]. Note that, in this case, function $g(\cdot)$ is non-Lipschitz at the origin, i.e. $\lim_{z\to 0}\frac{g(z)}{z} = \mathcal{K} \to \infty$. From (25), the solution converges for all η satisfying

$$\frac{\eta u \lambda_n^2}{\kappa \lambda_2} < \frac{\overline{\xi}^\top \overline{\xi}}{\overline{\varphi}^\top \overline{\varphi}}.$$

For non-Lipschitz case, if $\|\mathbf{x}(k)-\mathbf{x}^*\|_2^2 \to 0$, we have

$$\frac{\overline{\xi}^\top \overline{\xi}}{\overline{\varphi}^\top \overline{\varphi}} = \frac{1}{\mathcal{K}^2} \to 0,$$

implying that *to reach the exact optimizer* $\mathbf{x}^*$ we need $\eta \to 0$. Next, given $0 < \eta < \overline{\eta}$ we want to know how close can we get to the optimizer $\mathbf{x}^*$ and optimal value F^*. Using (13) we have

$$\frac{\eta\lambda_n^2 \overline{\varphi}^\top \overline{\varphi}}{4u\kappa\lambda_2} \leq \frac{\xi^\top \xi}{4u^2} \leq \|\mathbf{x}(k)-\mathbf{x}^*\|_2^2. \tag{35}$$

Therefore, for given $\eta < \overline{\eta}$,

$$0 < \frac{\eta\lambda_n^2}{4u\kappa\lambda_2} < \frac{\|\mathbf{x}(k)-\mathbf{x}^*\|_2^2}{\overline{\varphi}^\top \overline{\varphi}}. \tag{36}$$

This means that we cannot get arbitrarily close to the optimizer. Recall that in the case for which the nonlinearity is Lipschitz, as $\|\mathbf{x}(k)-\mathbf{x}^*\|_2^2 < \varepsilon \to 0$ the RHS of (36) always remains greater than $\frac{1}{16u^2\mathcal{K}^2}$ which is satisfied for $0 < \eta < \overline{\eta}$ via (23) (as shown in the proof of Theorem 1). However, for non-Lipschitz mapping $g(\cdot)$, the RHS $\to 0$ and therefore the inequality cannot be satisfied for $\varepsilon \to 0$ as the LHS is a positive number, and steady-state non-zero residual follows (35). For continuous-time dynamics, however, the convergence to the optimizer can be proved, e.g., see the results in [27], [48].

iv) The upper-bound $\overline{\eta}$ on the step-rate inversely depends on the Lipschitz constant u of the objective function f_i. For a fixed u, a larger value of $v \leq u$ implies tighter bound on the convergence rate in RHS of (26). For quadratic cost as in (41) (with $u = v$) we get the tightest bound on the convergence rate.

v) For the nonlinear function g, the ratio $\frac{\kappa}{\mathcal{K}^2} < 1$ appears in (23), while in (26) the gap between κ and $\mathcal{K}^2$ affects the convergence rate. From Fig. 1, coarser quantization (larger μ) implies smaller $\kappa = 1-\frac{\mu}{2}$ and larger $\mathcal{K} = 1+\frac{\mu}{2}$. This implies a tighter bound on η in (23) and a looser bound on the convergence rate in (26). On the other hand, as given by (29), coarser quantization results in higher ε-bound and possibly larger steady-state residual. See the simulations for better illustration.

vi) In presence of time-delays, feasible initialization can be done via, for example, using [10, Algorithm 2] over a longer time-scale as in Section A. This finite-time algorithm works irrespective of discrete (quantized) or real-valued information exchanges and gives many possible (quantized plus real) outputs.

V. SIMULATIONS

A. APPLICATION: OPTIMIZING THE CPU UTILIZATION

This application focuses on optimizing the CPU utilization across servers (computing nodes/servers) in data centers by carefully allocating CPU resources to workloads in a distributed fashion. The data centers are modelled as a set of $\mathcal{V}$ nodes. Each node $v_i \in \mathcal{V}$ can operate as a resource scheduler (which is a standard practice in modern data centers). The set of all jobs to be scheduled is $\mathcal{J}$. Each job $b_j \in \mathcal{J}$, (where $j \in \{1,\ldots,|\mathcal{J}|\}$) is the group of tasks and requires ρ_j cycles to be executed. The amount of ρ_j cycles required for job b_j to be executed is known before the optimization operation. At each node v_i, the total workload due to arriving jobs is denoted by l_i. Furthermore, the time period for which the optimization runs the jobs on the servers (before the next optimization operation for a new set of resource allocation) is defined as T_h. At each node v_i, the CPU capacity during the optimization operation is equal to $\pi_i^{\max} := c_i T_h$, where c_i is the sum of all clock rate frequencies of all processing cores of node v_i given in cycles/second. For each node v_i, the CPU availability at optimization step m[6] (i.e., at time step

[6]The CPU optimization step m is different from the time-index k of the algorithm. In fact, the algorithm runs between every two consecutive optimization steps m and $m+1$.

mT_h with T_h as the time period of the optimization operation) is $\pi_i^{\text{avail}}[m] := \pi_i^{\max} - u_i[m]$, where $u_i[m]$ is the number of unavailable/occupied cycles due to predicted utilization from already running tasks over the time horizon T_h at step m.

Denote the total amount of resources demanded at a specific optimization step m as $\rho[m] := \sum_{b_j[m]\in\mathcal{J}[m]} \rho_j[m]$. Furthermore, denote the total available capacity as $\pi^{\text{avail}}[m] := \sum_{v_i\in\mathcal{V}} \pi_i^{\text{avail}}[m]$. We have that T_h at step m, is chosen such that $\rho[m] \le \pi^{\text{avail}}[m]$, i.e., the total amount of resources demanded meets the total available resources. This indicates that the demand does not exceed the available resources in the network. T_h can be chosen appropriately to fulfill this (as the box constraints). Each node needs to calculate the optimal solution at every *optimization step m* by executing a distributed algorithm (with *time step k*). The exchanged information over the network (and the allocated CPU workloads are quantized). The algorithm can take into account such nonlinearities. In [1] every node balances its CPU such that each node utilizes the same percentage of its own capacity (under the feasibility constraint). This *balancing strategy* calculates the *optimal* workload $w_i^*[m]$ to be received at optimization step m such that $\forall v_i, v_j \in \mathcal{V}$

$$\frac{w_i^*[m] + u_i[m]}{\pi_i^{\max}} = \frac{w_j^*[m] + u_j[m]}{\pi_j^{\max}} = \frac{\rho[m] + u_{\text{tot}}[m]}{\pi^{\max}}, \quad (37)$$

where $\pi^{\max} := \sum_{v_i\in\mathcal{V}} \pi_i^{\max}$ and $u_{\text{tot}}[m] = \sum_{v_i\in\mathcal{V}} u_i[m]$. *Note that in the remainder we drop the index m, since we consider a single optimization step.* Each node maintains a scalar quadratic local cost function of the form,

$$f_i(z) = \frac{1}{2}\alpha_i(z - \rho_i)^2, \quad (38)$$

with $\alpha_i > 0$, $\rho_i \in \mathbb{R}$ as the positive demand at node v_i, and z as a global optimization parameter (that determines the optimal workload at each node). Each node needs to calculate the optimal parameter $z^* \in \mathcal{Z}$ such that $z^* = \arg\min_{z\in\mathcal{Z}} \sum_{v_i\in\mathcal{V}} f_i(z)$, where $\mathcal{Z}$ denotes the set of all feasible values of z, e.g., as the box constraints $\underline{m}_i \le z_i \le \overline{M}_i$. Its closed form solution for the quadratic cost (38) is

$$z^* = \frac{\sum_{v_i\in\mathcal{V}} \alpha_i\rho_i}{\sum_{v_i\in\mathcal{V}} \alpha_i}. \quad (39)$$

From [1], in order to calculate the optimal *balancing* workload according to (37), we need the solution of z^* to be

$$z^* = \frac{\sum_{v_i\in\mathcal{V}} \pi_i^{\max} \frac{\rho_i+u_i}{\pi_i^{\max}}}{\sum_{v_i\in\mathcal{V}} \pi_i^{\max}} = \frac{\rho + u_{\text{tot}}}{\pi^{\max}}. \quad (40)$$

From (40) we modify (38) as

$$f_i(z) = \frac{1}{2}\pi_i^{\max}\left(z - \frac{\rho_i + u_i}{\pi_i^{\max}}\right)^2. \quad (41)$$

This means that each node (i) computes its proportion of workload, and (ii) from its workload proportion it can calculate to receive the optimal workload w_i^* equal to

$$w_i^* = \frac{\rho + u_{\text{tot}}}{\pi^{\max}}\pi_i^{\max} - u_i. \quad (42)$$

Recall, however, that the allocated workload by (42) gives the optimal allocation subject to the balancing constraint in (37). In other words, it is possible to reach lower CPU allocation costs by disregarding this balancing condition and considering more general cost models as

$$f_i(z_i) = \frac{1}{2}\alpha_i(z_i - \rho_i)^2, \quad (43)$$

where $z_i \neq z_j$, in general. Note the subtle difference here as the factors z_i in (43) could be unequal (compared to the same z in formulation (38)). Substituting w_i from (37),

$$f_i(w) = \frac{1}{2\pi_i^{\max}}(w_i - \rho_i)^2, \quad (44)$$

This convex formulation gives a lower cost by replacing the balancing constraint $\frac{w_i+\rho_i}{\pi_i^{\max}} = \frac{w_j+\rho_j}{\pi_j^{\max}}$ (or $z_i = z_j$) with general sum-preserving constraint $\sum_{i=1}^n w_i = \sum_{i=1}^n w_i^* = \rho$. Note that we assign the same amount of overall workloads as given by (40). The modified version of (41) is then

$$f_i(w_i) = \frac{1}{2\pi_i^{\max}}(w_i - \rho_i)^2 \text{ s.t.} \sum_{i=1}^n w_i = \rho, \ \mathbf{z} \in \mathcal{Z} \quad (45)$$

with $\mathbf{z} \in \mathcal{Z}$ as the box constraints. Note that this box constraint makes the solution non-trivial, in general. The formulation (45) reshapes the problem in the form $\mathcal{P}_1$. In general, the cost of workloads assigned by (41) is always less than (or equal to) the cost associated with the balanced model. We add some penalty functions to keep the servers at an operating point away from the capacity, in fact, below $70-80\%$ of their capacity (due to the uncertainty of the processing times) since the Mean Response Time of the servers grows (exponentially) at some point [54]. As a rule-of-thumb, we address this concern by box constraints on the load-to-capacity ratios as $0 \le z_i \le 0.75$ (with $z_i = \frac{w_i+\rho_i}{\pi_i^{\max}}$).

For numerical simulation, a network of $n = 12$ servers with the following parameters for cost function (41) is considered: $\rho_i \in [10\ 30]$, $u_i \in [10\ 40]$, $\pi_i^{\max} = 80$. The box constraints are $0 \le w_i \le 0.75\pi_i^{\max} - \rho_i = M_i$. For simulation initialization we set the w_i values for some random nodes (chosen by `randperm` command) equal to $\min(M_i)$, for one node equal to $\text{mod}(\rho, \min(M_i))$, and the rest equal to 0. The overall resources are then $\sum_{i=1}^n w_i = \rho$. The parameters values are: $\rho = 190$, $\eta = 0.5$, and $\mu = 0.0675$. The simulation is done over a self-damped directed cyclic graph with balanced link weights and $\lambda_2 = 0.134$, $\lambda_n = 2$. Substituting $\kappa = 1 - \frac{\mu}{2} = 0.9663$, $\mathcal{K} = 1 + \frac{\mu}{2} = 1.0337$, $u = \frac{1}{2\pi_i^{\max}} = 0.0063$ in (23) gives the max step rate $\overline{\eta} = 4.85$. Table 1 compares our optimal allocation with the balancing strategy in [1].

TABLE 1 **Comparison Between the Allocated Workload Costs for the Same Amount of Overall Workload ρ: The Cost Given by (45) Versus Balancing Model in [1]**

Cost Model	$\rho = 190$	$\rho = 235$	$\rho = 250$
Allocation via Eq. (45)	0.17	0.96	0.18
Balancing via Ref. [1]	2.81	3.39	3.58

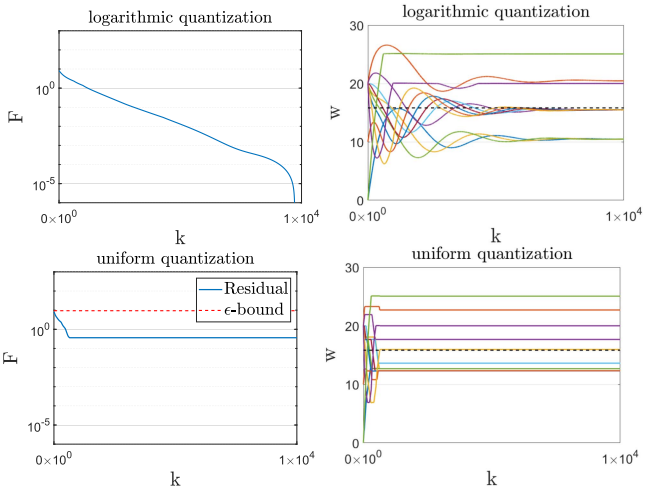

FIGURE 2. **CPU scheduling under two quantization approaches: (left) The residual under logarithmic quantization as a strongly sign-preserving nonlinearity is always decreasing and converges to zero, while for uniform quantizer only convergence to the ε-neighborhood (shown by dashed red) is guaranteed. (right) under both approaches, the solutions remain anytime-feasible and the averages of workloads (shown by black dashed lines) remain constant.**

TABLE 2 **Comparison on the Elapsed-Time of the Quantized (14) With the Linear Solution (Over 10^4 Iterations)**

Solution	linear	uni quant	log quant
Run time (sec)	0.13	0.12	0.39

The residuals $\overline{F} = F(\mathbf{x}) - F^*$ as the Lyapunov function under the two quantization schemes are compared in Fig. 2, which is decreasing towards zero under the logarithmic quantization for $\eta < \overline{\eta}$. The residual ε-accuracy bound from (29) is equal to $\overline{\varepsilon} = 9.3$; this implies the worst performance under uniform quantization. The average of states (black dashed lines) are constant, ensuring all-time feasibility. In addition, to give an idea on the computational complexity of the proposed solution (14) (with quantized nonlinearities), Table 2 compares the running time of the algorithm in MATLAB R2021b Intel Core i5 @ 2.4 GHz processor 8 GB RAM using the `tic-toc` functions.

Over the same setup and parameters, CPU allocation under time delays over the data-transmission network is given in Fig. 3. We run the simulation over the weight-symmetric cyclic graph (with $\lambda_2 = 0.09$, $\lambda_n = 1.33$) in Section IV-B assuming known but random delays. From (23) the (sufficient) bound on the step-rate is $\overline{\eta} = 7.26$ for no latency and, from Theorem 3 in the presence of delays, $\eta(\overline{\tau} + 1) < 7.26$. The

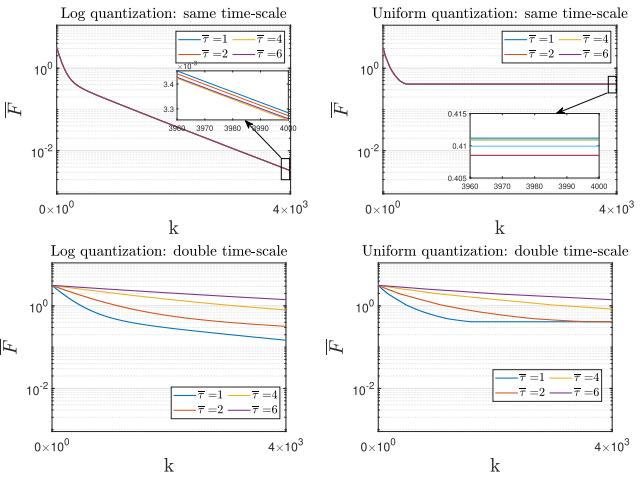

FIGURE 3. **CPU allocation under logarithmic and uniform quantization is shown subject to latency. The communication delays are assumed random, heterogeneous, but known and bounded by max delay $\overline{\tau} = 1, 2, 4, 6$ steps. (Top) under same time-scale scenario and (Bottom) under double time-scale scenario.**

simulations for both quantization schemes and both (same and double time-scale) delay scenarios with $\overline{\tau} = 1, 2, 4, 6$ and $\eta = 0.5$ are given in Fig. 3. The arbitrary delays in the range $[0\ \overline{\tau}]$ are generated via MATLAB `rand`. As it is clear from the figure, due to longer waiting time, the double time-scale scenario converges slower; however, from Theorem 2, it always converges for any $\overline{\tau}$ (for $\eta < 7.26$). In contrast the same time-scale scenario may not converge (for large $\overline{\tau}$) if $\eta(\overline{\tau} + 1) > 7.26$.

B. LOGARITHMIC PENALTY + NON-QUADRATIC COST: CONDITION NUMBER AND QUANTIZATION LEVEL

Recall that the proposed solution in this paper, to solve $\mathcal{P}_1$ based on the cost model (45), can optimize general non-quadratic cost functions, e.g., due to additive logarithmic or max-based penalty/barrier functions to the cost function (45). For this simulation, we consider non-quadratic cost as in [38],

$$\sum_{i=1}^{n} f_i(\mathbf{x}_i) = \sum_{i=1}^{n} \omega_i(\mathbf{x}_i - \alpha_i)^4 + \frac{\sigma}{\rho}\log(1 + \exp(\rho u)) \quad (46)$$

with cost parameters $b = 20$, random $\alpha_i \in [-5\ 5]$, random $\omega_i \in [0\ 0.5]$, $\sigma = 1$. The logarithmic penalty term with $\rho = 1$ is added to the objective function with weight factor $\sigma = 1$, $\underline{m}_i = 0$, $\overline{M}_i = 5$ (implying penalty terms $u_1 = [\mathbf{x}_i - 5]^+$, $u_2 = [0 - \mathbf{x}_i]^+$). The topology switches every 20 steps between 4 (disconnected) undirected network topologies while their union is an undirected connected cycle, i.e., Assumption 2 holds for $B = 80$. The simulation results are shown in Fig. 4 for $\eta = 0.0005$. The simulation is done for logarithmic and uniform quantization (with $\mu = 0.5$) as compared to the single-bit protocol [4] (with single bit of data exchange), protocol subject to saturation [27] with sat level equal to 1, classic linear solution [43], and signum-based solution for faster convergence [27], [31] with $g(z) = \text{sgn}^{\nu_1}(z) + \text{sgn}^{\nu_2}(z)$

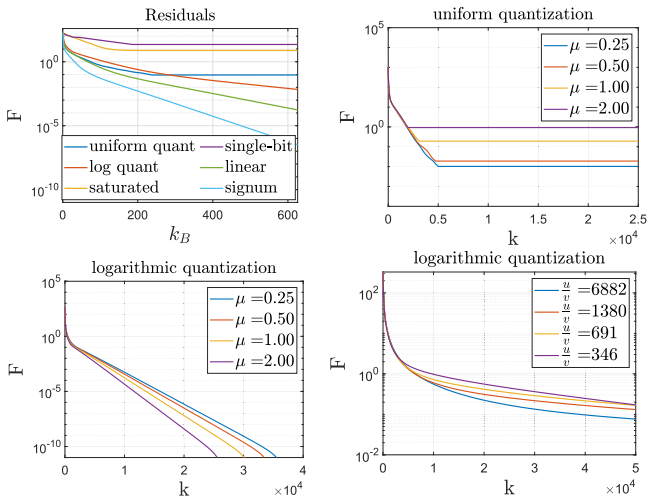

FIGURE 4. (TopLeft) Different allocation strategies to optimize non-quadratic cost model (46) over undirected cyclic networks changing every 20 steps and uniformly connected over every $B = 80$ steps. The horizontal axis shows the iterations over B time scale, i.e., $k_B = \lfloor \frac{k}{B} \rfloor$. Saturated [27], single-bit [4], linear [43], and signum-based [27], [31] solutions are also given for comparison. (TopRight) The steady-state residuals under different quantization levels. (BottomLeft) Convergence under different quantization levels μ. (BottomRight) Convergence under different condition numbers of the objective function (via parameter σ)

with $\mathrm{sgn}^{\nu}(z) := z\|z\|^{\nu-1}$ for $\nu_1 = 0.5$, $\nu_2 = 1.5$, since the sign-based nonlinearity is not upper-sector-bounded we consider small values of η to resemble continuous-time dynamics, see [27] for details. We further compared the residuals under different quantization levels μ for both logarithmic and uniform quantization. From the figure, for larger μ uniform quantization results in larger steady-state residual (Remark 3-(v) and (29)) while logarithmic quantization gives faster convergence. For the logarithmic quantization, we further compared the convergence rate for different condition numbers by tuning σ. The simulation parameters are: $b = 10$, random $\alpha_i \in [-5\ 5]$, random $\omega_i \in [0\ 0.5]$, $\mu = 1$ (i.e., $\kappa = 0.5$ and $\mathcal{K} = 1.5$), $\underline{m}_i = 0$, $\overline{M}_i = 5$, $\eta = 0.0005$. The network is considered as an undirected cycle with $\lambda_2 = 0.38$, $\lambda_n = 4$. We change the factor as $\sigma = [0.05, 0.25, 0.5, 1]$ which results in different condition numbers $\frac{u}{v}$ shown in the figure. In this example, for larger condition numbers the convergence is faster.

VI. CONCLUSION

The optimal allocation of resources over a weight-balanced directed network is addressed. Our nonlinear solution can provide quantized coordination with resource-demand feasibility at all times. The solution advances the state-of-the-art by simultaneously addressing (i) anytime-feasibility, (ii) quantization and ε-accuracy, (iii) latency, and (iv) uniform-connectivity. The results, therefore, allow for the design of algorithms for limited (or more cost-efficient) bandwidth by proper quantization over the dynamic networks (with intermittent connectivity). Overall, the solution is applicable in more general nonlinear setups to address convergence rate and robustness in future directions. Further, the weight-balance and uniform-connectivity assumption (in contrast to weight-stochasticity and all-time connectivity) allow for convergence under link removal and packet drops over switching networks, which is worth investigating.

Recall that, as discussed in [18] and our previous work [27], the sum-preserving problem formulation (2) (and (1)) can be extended to $\mathbf{y}_i \in \mathbb{R}^m, \mathbf{b} \in \mathbb{R}^p$ with $p, m > 1$ in general and a coupling-constraint in the form $\sum_{i=1}^n A_i \mathbf{y}_i = \mathbf{b}$ with $A_i \in \mathbb{R}^{p\times m}$. Using the notion of *slack variables*, it is possible to address *inequality* coupling-constraints $\sum_{i=1}^n A_i \mathbf{y}_i \preceq \mathbf{b}$ as in [23], [55]. This is by defining n additional (auxiliary) slack variables $\mathbf{s}_i \in \mathbb{R}^p$ such that $\sum_{i=1}^n (A_i \mathbf{y}_i + \mathbf{s}_i) = \mathbf{b}$ with $0 \preceq \mathbf{s}_i \preceq \bar{\mathbf{s}}$ as extra box constraints, see [23, Eq. (16)]. Each node (and local constraint) is associated with one slack variable, and the new state changes to $\widetilde{\mathbf{y}} = [\mathbf{y}; \mathbf{s}]$ with new constraint $\sum_{i=1}^n \widetilde{A}_i \widetilde{\mathbf{y}}_i = \mathbf{b}$ and $\widetilde{A}_i := [A_i\ I_p] \in \mathbb{R}^{p\times(m+p)}$. Certain convexity and connectivity assumptions need to be addressed for this formulation; see details in [23], [55]. This is one direction of our future research.

ACKNOWLEDGMENT

The authors would like to thank Prof. Jorge Cortes and Prof. Usman A. Khan for fruitful discussions.

REFERENCES

[1] A. I. Rikos, A. Grammenos, E. Kalyvianaki, C. N. Hadjicostis, T. Charalambous, and K. H. Johansson, "Optimal CPU scheduling in data centers via a finite-time distributed quantized coordination mechanism," in *Proc. 60th IEEE Conf. Decis. Control*, 2021, pp. 6276–6281.

[2] Z. Liu and O. Stursberg, "Distributed optimization for mixed-integer consensus in multi-agent networks," in *Proc. Eur. Control Conf.*, 2022, pp. 2157–2163.

[3] M. O. Sayin and S. S. Kozat, "Compressive diffusion strategies over distributed networks for reduced communication load," *IEEE Trans. Signal Process.*, vol. 62, no. 20, pp. 5308–5323, Oct. 2014.

[4] M. Doostmohammadian, "Single-bit consensus with finite-time convergence: Theory and applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 4, pp. 3332–3338, Aug. 2020.

[5] T. T. Doan and C. L. Beck, "Distributed resource allocation over dynamic networks with uncertainty," *IEEE Trans. Autom. Control*, vol. 66, no. 9, pp. 4378–4384, Sep. 2021.

[6] S. S. Kia, J. Cortés, and S. Martínez, "Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication," *Automatica*, vol. 55, pp. 254–264, 2015.

[7] U. A. Khan, W. U. Bajwa, A. Nedić, M. G. Rabbat, and A. H. Sayed, "Optimization for data-driven learning and control," *Proc. IEEE*, vol. 108, no. 11, pp. 1863–1868, Nov. 2020.

[8] R. Xin, S. Kar, and U. A. Khan, "Decentralized stochastic optimization and machine learning: A unified variance-reduction framework for robust performance and fast convergence," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 102–113, May 2020.

[9] H. Sayyaadi and M. Moarref, "A distributed algorithm for proportional task allocation in networks of mobile agents," *IEEE Trans. Autom. Control*, vol. 56, no. 2, pp. 405–410, Feb. 2011.

[10] A. Cherukuri and J. Cortés, "Distributed generator coordination for initialization and anytime optimization in economic dispatch," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 3, pp. 226–237, Sep. 2015.

[11] C. Nowzari, V. M. Preciado, and G. J. Pappas, "Optimal resource allocation for control of networked epidemic models," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 2, pp. 159–169, Jun. 2017.

[12] B. Gharesifard and J. Cortés, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 781–786, Mar. 2014.

[13] B. Can, S. Soori, M. M. Dehnavi, and M. Gürbüzbalaban, "L-DQN: An asynchronous limited-memory distributed Quasi-Newton method," in *Proc. IEEE Conf. Decis. Control*, 2021, pp. 2386–2393.
[14] D. Wang, Z. Wang, M. Chen, and W. Wang, "Distributed optimization for multi-agent systems with constraints set and communication time-delay over a directed graph," *Inf. Sci.*, vol. 438, pp. 1–14, 2018.
[15] C. M. de Galland, R. Vizuete, J. M. Hendrickx, P. Frasca, and E. Panteley, "Random coordinate descent algorithm for open multi-agent systems with complete topology and homogeneous agents," in *Proc. IEEE Conf. Decis. Control*, 2021, pp. 1701–1708.
[16] F. Shahriari-Mehr, D. Bosch, and A. Panahi, "Decentralized constrained optimization: Double averaging and gradient projection," in *Proc. IEEE Conf. Decis. Control*, 2021, pp. 2400–2406.
[17] H. Reisizadeh, B. Touri, and S. Mohajer, "Adaptive bit allocation for communication efficient distributed optimization," in *Proc. IEEE Conf. Decis. Control*, 2021, pp. 1994–2001.
[18] X. Wu, S. Magnusson, and M. Johansson, "A new family of feasible methods for distributed resource allocation," in *Proc. IEEE Conf. Decis. Control*, 2021, pp. 3355–3360.
[19] N. S. Aybat and E. Y. Hamedani, "Distributed primal-dual method for multi-agent sharing problem with conic constraints," in *Proc. 50th IEEE Asilomar Conf. Signals, Syst. Comput.*, 2016, pp. 777–782.
[20] A. Nedić, A. Olshevsky, and W. Shi, "Improved convergence rates for distributed resource allocation," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 172–177.
[21] G. Banjac, F. Rey, P. Goulart, and J. Lygeros, "Decentralized resource allocation via dual consensus ADMM," in *Proc. IEEE Amer. Control Conf.*, 2019, pp. 2789–2794.
[22] R. Carli and M. Dotoli, "Distributed alternating direction method of multipliers for linearly constrained optimization over a network," *IEEE Control Syst. Lett.*, vol. 4, no. 1, pp. 247–252, Jan. 2020.
[23] A. Falsone, I. Notarnicola, G. Notarstefano, and M. Prandini, "Tracking-ADMM for distributed constraint-coupled optimization," *Automatica*, vol. 117, 2020, Art. no. 108962.
[24] X.-F. Wang, Y. Hong, X.-M. Sun, and K.-Z Liu, "Distributed optimization for resource allocation problems under large delays," *IEEE Trans. Ind. Electron.*, vol. 66, no. 12, pp. 9448–9457, Dec. 2019.
[25] C. Enyioha, S. Magnússon, K. Heal, N. Li, C. Fischione, and V. Tarokh, "On variability of renewable energy and online power allocation," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 451–462, Jan. 2018.
[26] M. Vrakopoulou, B. Li, and J. L. Mathieu, "Chance constrained reserve scheduling using uncertain controllable loads Part I: Formulation and scenario-based analysis," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 1608–1617, Mar. 2019.
[27] M. Doostmohammadian, A. Aghasi, M. Vrakopoulou, and T. Charalambous, "1st-order dynamics on nonlinear agents for resource allocation over uniformly-connected networks," in *Proc. IEEE Conf. Control Technol. Appl.*, 2022.
[28] Y. Zhu, W. Ren, W. Yu, and G. Wen, "Distributed resource allocation over directed graphs via continuous-time algorithms," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 2, pp. 1097–1106, Feb. 2021.
[29] T. Anderson, C. Chang, and S. Martínez, "Distributed approximate newton algorithms and weight design for constrained optimization," *Automatica*, vol. 109, 2019, Art. no. 108538.
[30] C. Xi, R. Xin, and U. A. Khan, "ADD-OPT: Accelerated distributed directed optimization," *IEEE Trans. Autom. Control*, vol. 63, no. 5, pp. 1329–1339, May 2018.
[31] S. S. Stanković, M. Beko, and M. S. Stanković, "Robust nonlinear consensus seeking," in *Proc. 58th IEEE Conf. Decis. Control*, 2019, pp. 4465–4470.
[32] C. N. Hadjicostis and T. Charalambous, "Average consensus in the presence of delays in directed graph topologies," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 763–768, Mar. 2014.
[33] D. F. Coutinho, M. Fu, and C. E. de Souza, "Input and output quantized feedback linear systems," *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 761–766, Mar. 2010.
[34] R. Xin, D. Jakovetić, and U. A. Khan, "Distributed Nesterov gradient methods over arbitrary graphs," *IEEE Signal Process. Lett.*, vol. 26, no. 8, pp. 1247–1251, Aug. 2019.
[35] D. P. Bertsekas, "Necessary and sufficient conditions for a penalty method to be exact," *Math. Program.*, vol. 9, no. 1, pp. 87–99, 1975.
[36] D. P. Bertsekas, *Nonlinear Programming*, Belmont, MA, USA: Athena Scientific, 1997.
[37] S. P. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge, U.K.: Cambridge Univ. Press, 2004.
[38] T. T. Doan and A. Olshevsky, "Distributed resource allocation on dynamic networks in quadratic time," *Syst. Control Lett.*, vol. 99, pp. 57–63, 2017.
[39] S. Magnússon, C. Enyioha, N. Li, C. Fischione, and V. Tarokh, "Communication complexity of dual decomposition methods for distributed resource allocation optimization," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 4, pp. 717–732, Aug. 2018.
[40] M. Belabbas and X. Chen, "On integer balancing of directed graphs," *Syst. Control Lett.*, vol. 154, 2021, Art. no. 104980.
[41] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.
[42] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar, "Convexity, duality, and lagrange multipliers," in *Lecture Notes*, Cambridge, MA, USA: MIT Press, 2001.
[43] L. Xiao and S. Boyd, "Optimal scaling of a gradient method for distributed resource allocation," *J. Optim. Theory Appl.*, vol. 129, no. 3, pp. 469–488, 2006.
[44] M. Maggs, S. G. O'Keefe, and D. Thiel, "Consensus clock synchronization for wireless sensor networks," *IEEE Sensors J.*, vol. 12, no. 6, pp. 2269–2277, Jun. 2012.
[45] X. Wang, D. Jeske, and E. Serpedin, "An overview of a class of clock synchronization algorithms for wireless sensor networks: A statistical signal processing perspective," *Algorithms*, vol. 8, no. 3, pp. 590–620, 2015.
[46] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The flooding time synchronization protocol," in *Proc. 2nd Int. Conf. Embedded Networked Sensor Syst.*, 2004, pp. 39–49.
[47] N. K. Kalantari and S. M. Ahadi, "A logarithmic quantization index modulation for perceptually better data hiding," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1504–1517, Jun. 2010.
[48] K. Garg, M. Baranwal, and D. Panagou, "A fixed-time convergent distributed algorithm for strongly convex functions in a time-varying network," in *Proc. IEEE Conf. Decis. Control*, 2020, pp. 4405–4410.
[49] D. K. Molzahn et al., "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2941–2962, Nov. 2017.
[50] S. Yang, S. Tan, and J. Xu, "Consensus based approach for economic dispatch problem in a smart grid," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4416–4426, Nov. 2013.
[51] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
[52] D. Jurafsky and J. Martin, *Speech and Language Processing*, Hoboken, NJ, USA: Prentice Hall, 2020.
[53] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "Distributed subgradient methods and quantization effects," in *Proc. 47th IEEE Conf. Decis. Control*, 2008, pp. 4177–4184.
[54] E. Makridis, K. Deliparaschos, E. Kalyvianaki, A. Zolotas, and T. Charalambous, "Robust dynamic CPU resource provisioning in virtualized servers," *IEEE Trans. Serv. Comput.*, vol. 15, no. 2, pp. 956–969, Mar./Apr. 2022.
[55] R. Vujanic, P. M. Esfahani, P. J. Goulart, S. Mariéthoz, and M. Morari, "A decomposition method for large scale MILPs, with performance guarantees and a power system application," *Automatica*, vol. 67, pp. 144–156, 2016.