IEEE Control Systems Letters paper presented at
2023 American Control Conference (ACC)
San Diego, CA, USA. May 31 - June 2, 2023

Distributed optimal resource allocation using transformed primal-dual method

Solmaz S. Kia, *Senior Member, IEEE*, Jingrong Wei and Long Chen

***Abstract*— We consider an in-network optimal resource allocation problem in which a group of agents interacting over a connected graph want to meet a demand while minimizing their collective cost. The contribution of this paper is to design a distributed continuous-time algorithm for this problem inspired by a recently developed first-order transformed primal-dual method. The solution applies to cluster-based setting where each agent may have a set of subagents, and its local cost is the sum of the cost of these subagents. The proposed algorithm guarantees an exponential convergence for strongly convex costs and asymptotic convergence for convex costs. Exponential convergence when the local cost functions are strongly convex is achieved even when the local gradients are only locally Lipschitz. For convex local cost functions, our algorithm guarantees asymptotic convergence to a point in the minimizer set. Through numerical examples, we show that our proposed algorithm delivers a faster convergence compared to existing distributed resource allocation algorithms.**

I. Introduction

The recent decade has seen a flurry of new research in designing and analyzing optimization algorithms for large-scale in-network optimal decision-making problems. These large-scale problems often involve either constrained or unconstrained optimization of the sum of objective functions associated with local data in individual data centers or in-network local objectives; see [1] for examples in power networks. In this setting, the gradients can be computed perhaps with reasonable cost, while the Hessians are still expensive, and even their computation and/or storage become infeasible. As such, in the past decade, we have witnessed a surge in the design of first-order gradient descent algorithms with parallel/decentralized/distributed structures that are intended to address large-scale in-network optimization problems with efficient computation/communication/storage cost; see, e.g., [2] for a survey of recent distributed optimization algorithms. However, in many of the applications involving in-network optimizations, there is also a need for real-time adjustment of the system's response/decision to the present situation. Therefore, fast converging optimization algorithms are more and more in demand. As has been known in the classical optimization literature, improvement to the rate of convergence of optimization algorithms within a first-order framework can be obtained through methods such as quasi-Newton, heavy-ball, Nesterov, and other momentum methods [3]. However, these methods require storage of gradient at least one step behind. In this paper, we consider an in-network resource allocation problem and propose a first-order algorithm that only requires the agents' current gradient. This algorithm is based on a recently proposed transformed primal-dual [4] with proven fast convergence.

In an in-network optimal resource allocation problem, a group of N agents with limited resources wants to collectively meet a demand in a way that the overall cost consisted of sum of the local cost of the agents is minimized for the entire network, i.e.,

$$\begin{aligned} \mathsf{x}^{\star} = \operatorname{argmin}_{\mathbf{x}\in\mathbb{R}^N} \sum\nolimits_{i=1}^{N} f^i(x^i) \\ \text{subject to} \quad x^1 + x^2 + \cdots x^N = \mathsf{b}, \end{aligned} \tag{1}$$

where $x^i, i \in \mathcal{V} = \{1, 2, \cdots, N\}$ is the ith element of $\mathbf{x}$, the local cost functions $f^i : \mathbb{R} \to \mathbb{R}$ are convex differentiable and agents meet the demand $\mathsf{b} > 0$ in the equality constraint. When resources are bounded the inequality constraint $\underline{\mathsf{x}}^i \leq x^i \leq \bar{\mathsf{x}}^i$, $i \in \mathcal{V}$, should be added to the constraints of (1). Optimal in-network resource allocation appears in many optimal decision making tasks such as economic dispatch over power networks [5], [6], network resource allocation for wireless systems [7], [8] and optimal routing [9], [10].

Distributed solutions for problem (1) are studied extensively in the literature. For example, in the context of the power generator economic dispatch problem, [11]–[14] offer distributed solutions that solve a special case of (1) when local cost functions are quadratic. Distributed algorithm design with non-quadratic costs are presented in [15]–[17] in discrete-time form, and [18]–[23] in continuous-time form. These algorithms almost all are primal-dual solutions, some of them inspired by centralized solution of [24]. These results guarantee exponential convergence when the local costs are strongly convex with *globally* Lipschitz gradients. Only [21]–[23] guarantee convergence for convex functions. To induce robustness and also to yield convergence without strict convexity of the local cost functions, often augmented Lagrangian framework [25] is considered as in [16], [23].

In this paper, we propose a novel distributed continuous-time algorithm based on the newly proposed transformed primal-dual algorithm [4] to solve the optimal resource allocation problem (1). We propose our algorithm for a variation of in-network optimal resource allocation in which each agent

The first author is with the Department of Mechanical and Aerospace Engineering, University of California Irvine, Irvine, CA 92697. Her work was supported by NSF, under CAREER Award ECCS-1653838. The second and third authors are with the Department of Mathematics, University of California Irvine, Irvine, CA 92697. Their work was supported by NSF2012465. {solmaz,jingronw,lchen7}@uci.edu

$i \in \mathcal{V}$ may contain $n^i \geq 1$ sub agents with local costs $f^i_j(x^i_j)$. The optimal resource allocation objective then is

$$\begin{aligned} \mathsf{x}^\star = \mathrm{argmin}_{\mathbf{x}\in\mathbb{R}^N} \sum\nolimits_{i=1}^{N} \sum\nolimits_{j=1}^{n^i} f^i_j(x^i_j) \\ \text{subject to} \quad \mathbf{1}^\top_{n^1}\mathbf{x}^1 + \mathbf{1}^\top_{n^2}\mathbf{x}^2 + \cdots \mathbf{1}^\top_{n^N}\mathbf{x}^N = \mathsf{b}, \end{aligned} \tag{2}$$

where $\mathbf{x}^i = [x^i_1, \cdots, x^i_{n^i}]$, and $\mathbf{1}_{n^i} \in \mathbb{R}^{n^i}$ is the vector of all ones with dimension $n^i \geq 1$. We only consider the case of equality constraints and assume that local bounded decision constraints can be addressed using penalty function method as in [23]. We show the exponential convergence of our algorithm if the local cost function are strongly convex and the gradients are only locally Lipschitz. For convex local cost functions, the convergence is guaranteed to a point in the optimizer set. We compare the numerical results with distributed algorithms based on primal-dual algorithms. Our solver reduces oscillation in the trajectories and shows faster convergence for strongly convex and convex cases.

II. Notation and Preliminaries

We let $\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$, denote the set of real, positive real, and nonnegative real numbers, respectively. Throughout the paper, the bold letter like $\mathbf{x} \in \mathbb{R}^N$ denotes the vectors.Where clear from the context, we skip the dimension index. We let $\mathbb{R}^n$ be the n-dimensional Hilbert space with l_2 inner product and Euclidean norm $\|\cdot\|$. For a positive definite matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ and a vector $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\|^2_{\mathbf{M}}$ denotes $\mathbf{x}^\top\mathbf{M}\mathbf{x}$. For any proper closed convex function $f : \mathbb{R}^n \to \mathbb{R}$, we say f is μ-strongly convex, $\mu \in \mathbb{R}_{>0}$, if f is differentiable and for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$, we have $f(\mathbf{x}_2) - f(\mathbf{x}_1) - \nabla f(\mathbf{x}_1)^\top(\mathbf{x}_2 - \mathbf{x}_1) \geqslant \frac{\mu}{2}\|\mathbf{x}_1 - \mathbf{x}_2\|^2$. We say ∇f is globally Lipschitz if there exists $L \in \mathbb{R}_{>0}$ such that for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$,

$$(\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2))^\top(\mathbf{x}_1 - \mathbf{x}_2) \leqslant L\|\mathbf{x}_1 - \mathbf{x}_2\|^2. \tag{3}$$

∇f is locally Lipschitz if for any $\mathbf{x} \in \mathbb{R}^n$ there exists a neighborhood $\mathbf{U}$ such that for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathbf{U}$, (3) holds for some $L \in \mathbb{R}_{>0}$.

We denote an undirected connected graph by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ where $\mathcal{V}$ is the vertex set with $|\mathcal{V}| = N$, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the edge set and $\mathbf{A} = [a_{ij}]$ is the symmetric $N \times N$ adjacency matrix with $a_{ij} > 0$ if $(i, j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. Recall that a graph is undirected if and only if $a_{ij} = a_{ji}$, and connected if there is a path from every vertex to every other vertex of the graph. We denote the Laplacian matrix of the graph $\mathbf{L} = [l_{ij}]$. The Laplacian matrix is given by $\mathbf{L} = \mathbf{D} - \mathbf{A}$ where $\mathbf{D} = \mathrm{Diag}(\mathsf{d}_1, \cdots, \mathsf{d}_N)$, where $\mathsf{d}_i = \sum_{j=1}^N a_{ij}$. Recall that $\mathbf{L}$ is a positive semidefinite matrix, which for an undirected connected graph satisfies $\mathrm{rank}\,\mathbf{L} = N - 1$. Moreover, zero is a simple eigenvalue of $\mathbf{L}$ and $\{\alpha\mathbf{1} : \alpha \in \mathbb{R}\}$ is the eigenspace corresponding to the zero eigenvalue. We denote the eigenvalues $\mathbf{L}$ by $\{\lambda_i\}_{i=1}^N$ where $\lambda_1 = 0$ and $\lambda_i \leq \lambda_j$ for $i \leq j$. We let $\mathbf{r} = \frac{1}{\sqrt{N}}\mathbf{1}$ and $\mathbf{R}$ be an $N \times (N-1)$ orthonormal matrix where its ith column is the normalized eigenvector of λ_{i+1}. We denote $\mathbf{T} = \begin{bmatrix}\mathbf{r} & \mathbf{R}\end{bmatrix}$ and $\mathbf{\Lambda} = \mathrm{Diag}(\lambda_1, \cdots, \lambda_N)$. Notice that $\mathbf{T}^\top\mathbf{T} = \mathbf{I}$ and $\mathbf{T}^\top\mathbf{L} = \mathbf{\Lambda}\mathbf{T}^\top$. Direct computation gives $\mathbf{R}^\top\mathbf{R} = \mathbf{I}_{N-1}$ and $\mathbf{R}\mathbf{R}^\top = \mathbf{I}_N - \mathbf{r}\mathbf{r}^\top$. Lastly, if $p^i \in \mathbb{R}$ is a variable of agent $i \in \mathcal{V} = \{1, \cdots, N\}$, the aggregated p^i's of the network is the vector $\mathbf{p} = [p^1, \cdots, p^N]^\top \in \mathbb{R}^N$.

III. Problem Statement

We consider the optimal resource allocation problem (2) over a network of N agents under the following assumptions. We define $\bar{N} = \sum_{i=1}^N n^i$, $\mathcal{V} = \{1, \cdots, N\}$ and $\mathcal{V}^i = \{1, \cdots, n^i\}$ for $i \in \mathcal{V}$.

Assumption 1: The information sharing graph $\mathcal{G}$ is undirected and connected.

Assumption 2: The local costs of each agent $i \in \mathcal{V}$ at each subagent $j \in \mathcal{V}^i$, $f^i_j(x^i_j)$, are continuously differentiable convex functions with locally Lipschitz continuous gradients. Moreover, the optimization problem (2) has a finite minimum value $f^\star = f(\mathbf{x}^\star)$.

Assumption 2 ensures the problem (2) has a finite minimizer in the feasible set. The Karush-Kuhn-Tucker (KKT) conditions give a set of necessary and sufficient conditions to characterize the solution set of the convex optimization problem (2) as follows. The proof can be founded in [26].

Lemma 3.1 (KKT condition for convex optimization problems): *Let Assumption 2 hold. A point $\mathsf{x}^\star \in \mathbb{R}^{\bar{N}}$ is a solution of (2) iff there exists a $\mathsf{y}^\star \in \mathbb{R}$, such that $(\mathsf{x}^\star, \mathsf{y}) \in \mathcal{K}$, where*

$$\begin{aligned} \mathcal{K} = \{(\mathbf{x}, y) \in \mathbb{R}^{\bar{N}} \times \mathbb{R} \,|\, \nabla f(\mathbf{x}) + \mathbf{1}_{\bar{N}} y = 0, \\ \mathbf{1}^\top_{n^1}\mathbf{x}^1 + \cdots + \mathbf{1}^\top_{n^N}\mathbf{x}^N = \mathsf{b}\}. \end{aligned} \tag{4}$$

When the local costs f^i_j are all strongly convex, the KKT condition (4) has a unique solution.

Denote the Lagrangian of problem (2) by $\mathcal{L} = \sum_{i=1}^N \sum_j^{n^i} f^i_j(x^i_j) + y(\mathbf{1}^\top_{n^1}\mathbf{x}^1 + \cdots + \mathbf{1}^\top_{n^N}\mathbf{x}^N - \mathsf{b})$ and its Augmented Lagrangian by $\mathcal{L}_{\text{aug}} = \mathcal{L} + \frac{\rho}{2}\|\mathbf{1}^\top_{n^1}\mathbf{x}^1 + \cdots + \mathbf{1}^\top_{n^N}\mathbf{x}^N - \mathsf{b}\|^2$, $\rho > 0$. A well-known centralized solution for problem (1) is the saddle point dynamics [24]

$$\begin{aligned} \dot{x}^i_j &= -\frac{\partial \mathcal{L}}{\partial x^i_j} = -\nabla f^i_j(x^i_j) - y, \\ \dot{y} &= \frac{\partial \mathcal{L}}{\partial y} = \sum\nolimits_{k=1}^{N} \sum\nolimits_{l=1}^{n^k} x^l_k - \mathsf{b}, \end{aligned}$$

for $i \in \mathcal{V}$ and $j \in \mathcal{V}^i$. This solver has convergence guarantee only when every cost function $f^i_j(x^i_j)$ is strictly convex, see [27, Appendix B]. Using Augmented Lagrangian, convergence can be extended to convex cost by implementing

$$\dot{x}^i_j = -\frac{\partial \mathcal{L}_{\text{aug}}}{\partial x^i_j} = -\nabla f^i_j(x^i_j) - \rho(\sum_{k=1}^{N}\sum_{l=1}^{n^k} x^l_k - \mathsf{b}) - y, \tag{5a}$$

$$\dot{y} = \frac{\partial \mathcal{L}_{\text{aug}}}{\partial y} = \sum\nolimits_{k=1}^{N} \sum\nolimits_{l=1}^{n^k} x^l_k - \mathsf{b}, \tag{5b}$$

for $i \in \mathcal{V}$ and $j \in \mathcal{V}^i$. Some of the existing distributed algorithms, e.g., [18]–[23], for the optimal resource allocation problem are inspired by the aforementioned centralized solutions. The main premise of these approaches is to use a dynamic average consensus algorithm to track/generate the coupling term $\sum_{i=1}^{N}\sum_{j=1}^{n^i} x_i^j - \mathsf{b}$ in a distributed manner, most often for $n^i = 1$, $i \in \mathcal{V}$. In a recent work in [4], authors propose a centralized *transformed* primal-dual (TPD) method

$$\dot{x}_j^i = -\frac{\partial \mathcal{L}}{\partial x_j^i} = -\nabla f^i(x_j^i) - y, \quad i \in \mathcal{V},\ j \in \mathcal{V}^i, \tag{6a}$$

$$\dot{y} = \frac{\partial \mathcal{L}}{\partial y} - \mathbf{1}^\top \frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \sum_{k=1}^{N}\sum_{l=1}^{n^k} x_l^k - \mathsf{b} - \sum_{l=1}^{N}\sum_{l=1}^{n^k}(\nabla f_l^k(x^k) + y), \tag{6b}$$

as an alternative centralized solver for problem (1). This solver has guaranteed convergence for convex functions and shows faster convergence than the primal-dual dynamics, especially when the saddle point system is not strongly concave with respect to the dual variable. Recall that the augmented Lagrangian method extends the convergence of the primal-dual solver (5) to convex cost functions by introducing the augmented term $\frac{\rho}{2}\|\mathbf{1}_{n^1}^\top \mathbf{x}^1 + \cdots + \mathbf{1}_{n^N}^\top \mathbf{x}^N - \mathsf{b}\|^2$ into Lagrangian, which results in the stabilizing term $-\rho x_j^i$ in the dynamics of $\dot{x}_j^i$. Alternatively, noticed that the (augmented) Lagrangian is only concave but not strongly concave with respect to the dual variable, we shall introduce the stabilizing term in the the dual dynamics. As the critical points of the dynamic should not be changed, one intuitive choice is to add the terms from the primal dynamic to $\dot{y}$. In this case, we get (6b) and the corresponding stabilizing term for the dual variable dynamic $\dot{y}$ is $-y$. Therefore, the transformed primal-dual dynamics recover the loss of strong concavity with respect to y in the (augmented) Lagrangian.

IV. Distributed primal-dual algorithm

Invoking Lemma (3.1), the set of minimizers and their corresponding Lagrange multiplier is given by

We propose the continuous-time distributed transformed primal-dual (D-TPD) algorithm

$$\dot{v}^i = \sum\nolimits_{j=1}^{N} a_{ij}(y^i - y^j), \tag{7a}$$

$$\dot{y}^i = \sum_{l=1}^{n^i} x_l^i - \bar{\mathsf{b}}^i - \sum_{l=1}^{n^i}(\nabla f_l^i(x_l^i) + y^i) - \sum\nolimits_{j=1}^{N} a_{ij}(y^i - y^j) - v^i, \tag{7b}$$

$$\dot{x}_l^i = -\nabla f_l^i(x_l^i) - y^i, \quad l \in \mathcal{V}^i, \tag{7c}$$

initialized at $y^i(0), x_j^i(0) \in \mathbb{R}$, $j \in \mathcal{V}^i$ and $v^i(0) = 0$, $i \in \mathcal{V}$, as a distributed solver for optimization problem (2). Here, $\bar{\mathsf{b}}^i$, $i \in \mathcal{V}$ are defined such that $\sum_{j=1}^{N} \bar{\mathsf{b}}^j = \mathsf{b}$; e.g., $\bar{\mathsf{b}}^i = \frac{\mathsf{b}}{N}$, $i \in \mathcal{V}$ when all agents know the demand, or $\bar{\mathsf{b}}^j = \mathsf{b}$ and $\bar{\mathsf{b}}^i = 0$ $i \in \mathcal{V}\backslash\{j\}$ when only agent j knows the demand.

Our proposed D-TPD algorithm (7) is inspired by the centralized TPD algorithm (6), in which every agent has a local copy of the dual variable y, driven by local component $\sum_l^{n^i} x_l^i - \bar{\mathsf{b}}^i - \sum_l^{n^i}(\nabla f_l^i(x_l^i) - y^i)$ of each agent followed by a proportional integral agreement feedback $-\sum_{j=1}^{N} a_{ij}(y^i - y^j) - v^i$ to make y^i of each agent to eventually converge to y dynamics (6b). Compared with primal-dual algorithms, the dynamic of y^i is transformed with the local private data $\nabla f_j^i(x_j^i)$ and y^i. This can be viewed as approximation of gradient flow to (6). The D-TPD algorithm (7)'s aggregate representation reads

$$\dot{\mathbf{v}} = \mathsf{L}\mathbf{y}, \tag{8a}$$

$$\dot{\mathbf{y}} = \mathbf{\Gamma}(\mathbf{x} - \nabla f(\mathbf{x})) - \bar{\mathbf{b}} - \mathbf{D}\mathbf{y} - \mathsf{L}\mathbf{y} - \mathbf{v}, \tag{8b}$$

$$\dot{\mathbf{x}} = -\nabla f(\mathbf{x}) - \mathbf{\Gamma}^\top \mathbf{y}, \tag{8c}$$

where $\mathbf{\Gamma} = \text{Diag}(\mathbf{1}_{n^1}^\top, \cdots, \mathbf{1}_{n^N}^\top)$ and $\mathbf{D} = \text{Diag}(n^1, \cdots, n^N)$. Consider the change of variable

$$\boldsymbol{\nu} = \mathbf{T}^\top(\mathbf{v} - (\mathbf{\Gamma}\hat{\mathbf{x}}^\star - \bar{\mathbf{b}})), \tag{9a}$$

$$\boldsymbol{\eta} = (\mathbf{y} - \hat{\mathsf{y}}^\star \mathbf{1}), \tag{9b}$$

$$\boldsymbol{\chi} = (\mathbf{x} - \hat{\mathbf{x}}^\star), \tag{9c}$$

where $(\hat{\mathbf{x}}^\star, \hat{\mathsf{y}}^\star) \in \mathcal{K}$ is an arbitrary solution of the KKT equation of problem (2) resource allocation problem. Recall that $\mathbf{r}^\top \mathsf{L} = 0$, and note that $\mathbf{T}^\top \mathsf{L} = \mathbf{T}^\top \mathsf{L}\mathbf{R}\mathbf{R}^\top$. Also, because $\mathbf{\Gamma}\mathbf{\Gamma}^\top = \mathbf{D}$, we have $\mathbf{\Gamma}(\nabla f(\hat{\mathbf{x}}^\star) + \mathbf{\Gamma}^\top \hat{\mathsf{y}}^\star \mathbf{1}) = \mathbf{\Gamma}\nabla f(\hat{\mathbf{x}}^\star) + \mathbf{D}\,\hat{\mathsf{y}}^\star \mathbf{1} = \mathbf{0}$. Denote $\mathsf{L}^+ = \mathbf{R}^\top \mathsf{L}\mathbf{R}$. Then, using (9a), (8) can be written in the equivalent form

$$\dot{\nu}_1 = 0, \tag{10a}$$

$$\dot{\boldsymbol{\nu}}_{2:N} = \mathsf{L}^+ \mathbf{R}^\top \boldsymbol{\eta}, \tag{10b}$$

$$\dot{\boldsymbol{\eta}} = \mathbf{\Gamma}\boldsymbol{\chi} - \mathbf{\Gamma}(\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}^\star)) - \mathbf{D}\boldsymbol{\eta} - \mathsf{L}\boldsymbol{\eta} - \mathbf{R}\,\boldsymbol{\nu}_{2:N}, \tag{10c}$$

$$\dot{\boldsymbol{\chi}} = -(\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}^\star)) - \mathbf{\Gamma}^\top \boldsymbol{\eta}. \tag{10d}$$

For convenience in analysis we will study convergence of the equivalent representation (10). Let $(\bar{\boldsymbol{\nu}}_{2:N}, \bar{\boldsymbol{\eta}}, \bar{\boldsymbol{\chi}})$ denote an equilibrium point of (10b)-(10d), i.e.,

$$\mathbf{0} = \mathsf{L}^+ \mathbf{R}^\top \bar{\boldsymbol{\eta}},$$
$$\mathbf{0} = \mathbf{\Gamma}\bar{\boldsymbol{\chi}} - \mathbf{\Gamma}(\nabla f(\bar{\mathbf{x}}) - \nabla f(\hat{\mathbf{x}}^\star)) - \mathbf{D}\bar{\boldsymbol{\eta}} - \mathsf{L}\bar{\boldsymbol{\eta}} - \mathbf{R}\,\bar{\boldsymbol{\nu}}_{2:N},$$
$$\mathbf{0} = -(\nabla f(\bar{\mathbf{x}}) - \nabla f(\hat{\mathbf{x}}^\star)) - \mathbf{\Gamma}^\top \bar{\boldsymbol{\eta}},$$

where $\bar{\mathbf{x}} = \bar{\boldsymbol{\chi}} + \hat{\mathbf{x}}^\star$. Since L^+ is invertible and we have $\mathbf{1}^\top \mathsf{L} = \mathbf{0}^\top$, $\mathbf{R}^\top \mathbf{1} = \mathbf{0}$, $\mathbf{1}^\top \mathbf{R} = \mathbf{0}^\top$, it is straightforward to confirm that the set of equilibrium points of (10b)-(10d) are

$$\mathcal{S} = \{(\bar{\boldsymbol{\nu}}_{2:N}, \bar{\boldsymbol{\eta}}, \bar{\boldsymbol{\chi}}) \in \mathbb{R}^{N-1} \times \mathbb{R}^N \times \mathbb{R}^{\bar{N}} | \bar{\boldsymbol{\chi}} = \mathbf{x}^\star - \hat{\mathbf{x}}^\star,$$
$$\bar{\boldsymbol{\eta}} = \mathsf{y}^\star \mathbf{1} - \hat{\mathsf{y}}^\star \mathbf{1},\ \bar{\boldsymbol{\nu}}_{2:N} = \mathbf{R}^\top \mathbf{\Gamma}\bar{\boldsymbol{\chi}},\ \ \forall (\mathbf{x}^\star, \mathsf{y}^\star) \in \mathcal{K}\}. \tag{11}$$

With the preliminaries in order, we are ready to make the first statement about the convergence guarantee of algorithm 7.

Theorem 4.1 (Convergence when the local cost functions are convex): *Let Assumption 1 and Assumption 2 hold. Starting from any initial condition $x_j^i(0) \in \mathbb{R}$, $j \in \mathcal{V}^i$, $y^i(0) \in \mathbb{R}$ and $v^i(0) = 0$, the trajectory of algorithm* (7) *for each agent $i \in \mathcal{V}$ to satisfy $(x_j^i, y^i) \to (\mathsf{x}_j^{i\star}, \mathsf{y}^\star)$ as $t \to \infty$, where $(\mathbf{x}^\star, \mathsf{y}^\star) \in \mathcal{K}$ where $\mathcal{K}$ is given in* (4).

Proof: Consider the equivalent representation (10) of the algorithm (7). To analyze stability of (10b)-(10d) consider the radially unbounded Lyapunov function

$$V_c = \frac{1}{2}\boldsymbol{\chi}^\top\boldsymbol{\chi} + \frac{1}{2}\boldsymbol{\eta}^\top\boldsymbol{\eta} + \frac{1}{2}\boldsymbol{\nu}_{2:N}^\top(\mathsf{L}^+)^{-1}\boldsymbol{\nu}_{2:N} + \sum\nolimits_{i=1}^{N}\sum\nolimits_{j=1}^{n^i}(f_j^i(x_j^i) - f_j^i(\hat{\mathsf{x}}_j^{i\star})) - \nabla f(\hat{\mathbf{x}}^\star)^\top\boldsymbol{\chi}. \quad (12)$$

Along the system trajectories of (10b)-(10d) we get $\dot{V}_c = -\boldsymbol{\eta}^\top\mathsf{L}\boldsymbol{\eta} - (\boldsymbol{\Gamma}^\top\boldsymbol{\eta} + (\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}^\star)))^\top(\boldsymbol{\Gamma}^\top\boldsymbol{\eta} + (\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}^\star))) - \boldsymbol{\chi}^\top(\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}^\star)) \leq 0$, where we invoked $\mathbf{D} = \boldsymbol{\Gamma}\boldsymbol{\Gamma}^\top$. $\dot{V}_c \leq 0$ follows from $\mathsf{L} \geq 0$ and the convexity of local cost functions, i.e., $\boldsymbol{\chi}^\top(\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}^\star)) \geq 0$. So far we have established that $\dot{V}_c \leq 0$. Let $\mathcal{S}_c = \{(\boldsymbol{\nu}_{2:N}, \boldsymbol{\eta}, \boldsymbol{\chi}) \in \mathbb{R}^{N-1} \times \mathbb{R}^N \times \mathbb{R}^{\bar{N}} | \dot{V}_c = 0\}$. The points in $\mathcal{S}_c$ satisfy

$$\boldsymbol{\Gamma}^\top\boldsymbol{\eta} + (\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}^\star))) = \mathbf{0}, \quad (13a)$$

$$\boldsymbol{\eta}^\top\mathsf{L}\boldsymbol{\eta} = \mathbf{0}, \quad (13b)$$

$$\boldsymbol{\chi}^\top(\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}^\star)) = \mathbf{0}. \quad (13c)$$

It follows from (13b) that for trajectories in $\mathcal{S}_c$ we have $\boldsymbol{\eta}$ in the span of $\mathbf{1}$. Then, using (13a), we obtain from (13c) that $\sum_{i=1}^{N}\sum_{j}^{n^i}\chi_j^i = 0$ or $\sum_{i=1}^{N}\sum_{j}^{n^i}x_j^i = \sum_{i=1}^{N}\sum_{j}^{n^i}\hat{\mathsf{x}}^{i\star} = \mathsf{b}$. Thus, the smallest invariant set of (10b)-(10d) in $\mathcal{S}_c$ is in fact $\mathcal{S}$ as in (11), the set of equilibrium points of (10b)-(10d). Then, it follows from the La Salle invariant set theorem [28, Theorem 3.4] that starting from every initial condition the trajectories of (10b)-(10d) converges to $\mathcal{S}$. Next, notice that (12) is radially unbounded Lyapunov function which is zero only at origin. Therefore, it follows from $\dot{V}_c \leq 0$ that origin is Lyapunov stable equilibrium point of (10b)-(10d). Since, $(\hat{\mathsf{x}}^\star, \hat{\mathsf{y}}^\star)$ used in (9) is an arbitrary point in $\mathcal{K}$, it is straightforward to argue that every other equilibrium point of (10b)-(10d) is Lyapunov stable. Therefore, it follows from [28, Theorem 4.20] that (10b)-(10d) is semistable, i.e., starting from any initial condition, the trajectories of (10b)-(10d) converge to one of its equilibrium points in $\mathcal{S}$. Then, given (9), as $t \to \infty$, D-TPD algorithm (7) under the stated initialization results in $(x_j^i(t), y^i(t)) \to (\mathsf{x}_j^{i\star}, \mathsf{y}^\star)$, $i \in \mathcal{V}$, $j \in \mathcal{V}^i$ where $(\mathsf{x}_j^{i\star}, \mathsf{y}^\star)$ is a point in $\mathcal{K}$. ∎

Next, we show that when the local costs are strongly convex the convergence guarantee of D-TPD algorithm is exponential. Notice that this guarantee does not require the customary global Lipschitezess of the gradient of the local costs.

Theorem 4.2 (Convergence when the local cost functions are strongly convex): *Let Assumption 1 and Assumption 2 hold. Additionally, assume each f_j^i, $i \in \mathcal{V}$ and $j \in \mathcal{V}^i$ is m_j^i-strongly convex. Starting from any initial condition $x_j^i(0) \in \mathbb{R}$, $j \in \mathcal{V}^i$, $y^i(0) \in \mathbb{R}$ and $v^i(0) = 0$, algorithm (7) result in $(x_j^i, y^i) \to (\mathsf{x}_j^{i\star}, \mathsf{y}^\star)$ exponentially fast as $t \to \infty$. Here, $(\mathbf{x}^{i\star}, \mathsf{y}^\star)$ is the unique solution of the KKT condition (4).*

Proof: Consider (10), the equivalent representation of (8). Consider the radially unbounded Lyapunov function $V = V_c + V_s$, where $V_s = \frac{\phi}{2}(\boldsymbol{\eta} + \mathbf{R}\boldsymbol{\nu}_{2:N})^\top\mathbf{D}^{-1}(\boldsymbol{\eta} + \mathbf{R}\boldsymbol{\nu}_{2:N}) + \frac{\phi}{2}\boldsymbol{\nu}_{2:N}^\top(\mathsf{L}^+)^{-1}\boldsymbol{\nu}_{2:N}$ for some $\phi \in \mathbb{R}_{>0}$ and V_c is given in (12). Note that $\dot{V}_s = \phi(\boldsymbol{\eta} + \mathbf{R}\boldsymbol{\nu}_{2:N})^\top\mathbf{D}^{-1}(\boldsymbol{\Gamma}\boldsymbol{\chi} - \boldsymbol{\Gamma}(\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}^\star)) - \mathbf{D}\boldsymbol{\eta} - \mathsf{L}\boldsymbol{\eta} - \mathbf{R}\,\boldsymbol{\nu}_{2:N} + \mathbf{R}\mathsf{L}^+\mathbf{R}^\top\boldsymbol{\eta}) + \phi\boldsymbol{\nu}_{2:N}^\top\mathbf{R}^\top\boldsymbol{\eta}$. Noting that $\mathbf{R}\mathsf{L}^+\mathbf{R}^\top = \mathsf{L}$, after some straightforward manipulations, we can write $\dot{V}_s = -\frac{\phi}{2}\|\boldsymbol{\eta} + \mathbf{R}\boldsymbol{\nu}_{2:N} - \boldsymbol{\Gamma}\boldsymbol{\chi} + \boldsymbol{\Gamma}(\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}^\star))\|_{\mathbf{D}^{-1}}^2 + \frac{\phi}{2}\|\boldsymbol{\Gamma}\boldsymbol{\chi} - \boldsymbol{\Gamma}(\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}^\star))\|_{\mathbf{D}^{-1}}^2 - \phi\boldsymbol{\eta}^\top(\mathbf{I} - \frac{1}{2}\mathbf{D}^{-1})\boldsymbol{\eta} - \frac{\phi}{2}\|\mathbf{R}\boldsymbol{\nu}_{2:N}\|_{\mathbf{D}^{-1}}^2$. Recall that by virtue of analysis in proof of Theorem 4.1 ($\dot{V}_c \leq 0$), we have established that the unique equilibrium point of (10b)-(10d) is Lyapunov stable. Therefore, starting from any initial condition trajectories of (10b)-(10d) are guaranteed to stay in a compact bounded set. Because local costs are differentiable and locally Lipschitz, then in that compact set, there is always an L for which we have $\|\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}^\star)\| \leq L\|\mathbf{x} - \hat{\mathbf{x}}^\star\|$. Thus, we can show from our derivation above that

$$\dot{V}_s \leq -\frac{\phi}{2}\|\boldsymbol{\eta} + \mathbf{R}\boldsymbol{\nu}_{2:N} - \boldsymbol{\Gamma}\boldsymbol{\chi} + \boldsymbol{\Gamma}(\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}^\star))\|_{\mathbf{D}^{-1}}^2 - \frac{\phi}{2}\|\mathbf{R}\boldsymbol{\nu}_{2:N}\|_{\mathbf{D}^{-1}}^2 + \phi\|\mathbf{D}^{-1}\|\|\boldsymbol{\Gamma}\|^2(1 + L^2)\|\boldsymbol{\chi}\|^2 - \phi\boldsymbol{\eta}^\top(\mathbf{I} - \frac{1}{2}\mathbf{D}^{-1})\boldsymbol{\eta}.$$

By invoking strong convexity of the local costs f_j^i we can write $-\boldsymbol{\chi}^\top(\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}^\star)) \leq -m\|\boldsymbol{\chi}\|^2$, where $m = \min\{\{m_j^i\}_{j=1}^{n^i}\}_{i=1}^{N}$. Then, considering the bound established for $\dot{V}_c$ in the proof of Theorem 4.1, $\dot{V} = \dot{V}_c + \dot{V}_s$ along the trajectories of (10b)-(10d) satisfies

$$\dot{V} \leq -\phi\boldsymbol{\eta}^\top(\mathsf{I} - \frac{1}{2}\mathbf{D}^{-1})\boldsymbol{\eta} - (m - \phi\|\mathbf{D}^{-1}\|\|\boldsymbol{\Gamma}\|^2(1 + L^2))\|\boldsymbol{\chi}\|^2 - \frac{\phi}{2}\|\boldsymbol{\eta} + \mathbf{R}\boldsymbol{\nu}_{2:N} - \boldsymbol{\Gamma}\boldsymbol{\chi} + \boldsymbol{\Gamma}(\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}^\star))\|_{\mathbf{D}^{-1}}^2 - \frac{\phi}{2}\|\mathbf{R}\boldsymbol{\nu}_{2:N}\|_{\mathbf{D}^{-1}}^2 - \boldsymbol{\eta}^\top\mathsf{L}\boldsymbol{\eta} - \|\boldsymbol{\Gamma}^\top\boldsymbol{\eta} + \nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}}^\star)\|^2.$$

Notice that $\mathsf{I} - \frac{1}{2}\mathbf{D}^{-1} > 0$. Also, since $\text{rank}(\mathbf{R}) = N - 1$, we have $\|\mathbf{R}\boldsymbol{\nu}_{2:N}\|_{\mathbf{D}^{-1}} = 0$ if and only if $\mathbf{v}_{2:N} = \mathbf{0}$. Next, note that there always exists a $\hat{\phi} > 0$ such that for $0 < \phi \leq \hat{\phi}$ we have $-(m - \phi\|\boldsymbol{\Gamma}\|^2(1 + L^2)) < 0$. Therefore, for any $0 < \phi \leq \hat{\phi}$, we have $\dot{V} < 0$. Similarly, we can establish a quadratic negative definite polynomial bound for $\dot{V}$. Therefore, we can establish an exponential convergence to origin for (10b)-(10d) by invoking [29, Theorem 4.10]. The exponential convergence of the D-TPD algorithm as stated in the theorem statement then follows from (9). ∎

V. Numerical Simulations

We demonstrate the performance of our algorithm via two examples. We compare the performance of our algorithm with that of the the following primal-dual based algorithms: the continuous-time distributed algorithm (PD) proposed in [21]; the initialization-free distributed projected (Proj) algorithms proposed in [20]; the initialization-free distributed coordination (dac+L∂) proposed in [6]; the distributed augmented Lagrangian algorithm (PD-AL) proposed in [23]. For both examples we consider our in-network resource allocation problems for a group of 6 agents interacting over an undirected connected graph depicted in Fig. 1. To compare

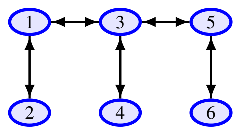

Fig. 1: A communication topology with edge weights of 1.

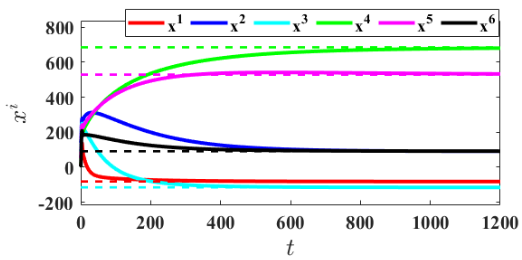

Fig. 2: Trajectories of the D-TPD dynamics for the IEEE 118 bus system. Horizontal dashed lines depict the centralized solution obtained using Matlab's constraint optimization solver 'fmincon'.

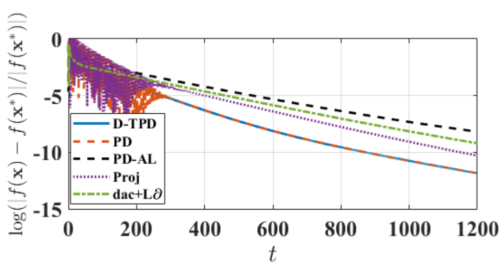

Fig. 3: Relative error of function value computed by D-TPD dynamics (blue) and PD (orange), PD-AL(black), Proj (purple), dac+L∂ (green).

with existing literature which do not consider subagents, we use $n^i = 1$, $i \in= \mathcal{V} = \{1, \cdots, 6\}$.

For first example, the local cost functions are $f^i(x^i) = a^i(x^i)^2 + b^i x^i + c^i$, where $(a^i, b^i, c^i), i \in \{1, 2, \cdots, 6\}$, which describe the cost for power generators in the IEEE 118 bus test model, located at buses $(4, 10, 18, 26, 54, 69)$. In this problem, the agents meet a demand $\mathsf{b} = 1200$ with their allocated powers. The optimization problem thus is

$$\mathbf{x}^\star = \text{argmin}_{\mathbf{x}\in\mathbb{R}^6} \sum\nolimits_{i=1}^{6} f^i(x^i)$$
$$\text{subject to} \quad x^1 + x^2 + x^3 + x^4 + x^5 + x^6 = 1200.$$

Figure 2 shows the trajectory of decision variables following D-TPD dynamics (7) and compares it to the centralized solution obtained by Matlab's constraint optimization solver 'fmincon'. As expected the decision variable x^i of each agent $i \in \{1, 2, 3, 4, 5, 6\}$ converges to its corresponding solution of the optimization problem.

The relative error of the function value on the trajectories computed by primal-dual based algorithms and D-TPD dynamics are shown in Fig. 3. Comparing rate of convergence of continuous-time algorithms is a subtle matter. To have a fair comparison, as proposed in [30], we assume that the decision variable evolution in each algorithm is of the form $\dot{x}^i(t) = u^i(t)$, where we think of $u^i(t)$ as control input. For example for the D-TPD algorithm $u^i(t) = -\nabla f^i(x^i) - y^i$.

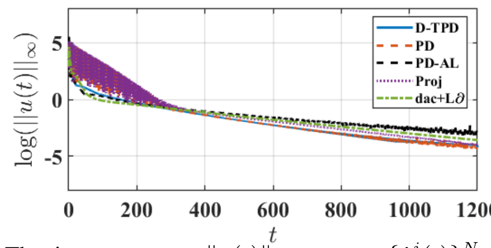

Fig. 4: The input energy $\|u(t)\|_\infty = \max\{\dot{x}^i(t)\}_{i=1}^N$ to the dynamic systems: D-TPD dynamics (blue) , PD (orange), PD-AL(black), Proj (purple), dac+L∂ (green).

We assume that an algorithm has accelerated convergence compared to another algorithm if it converges faster for almost the same control effort $\|u(t)\|_\infty = \max\{\dot{x}^i(t)\}_{i=1}^N$ as of the other algorithm. Notice that in a continuous-time algorithm a faster convergence can always be achieved by cranking up the 'input', i.e., using $\dot{x}^i = \alpha\, u^i$ for an $\alpha > 1$. Figure 4 shows the input size of the dynamic systems are at the comparable level, which means the acceleration of D-TPD that is observed in Fig. 3 is not due to scaling on the continuous-time system, for example by changing time scales. Compared with the PD algorithm, the D-TPD dynamics reduce the oscillation of the decision variables and the input energy to the system. That is, compared with other primal-dual based algorithms, the D-TPD algorithm accelerates the convergence of the decision variables without increasing the input size.

As a second example, consider

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}\in\mathbb{R}^6} \sum\nolimits_{i=1}^{6} f^i\left(x^i\right), \qquad (14)$$
$$\text{subject to} \quad x^1 + x^2 + x^3 + x^4 + x^5 + x^6 = 2.$$

where the local cost functions are

$$f^i\left(x^i\right) = \begin{cases} 0, & \left|x^i\right| \le \beta^i \\ \dfrac{1}{2\alpha^i}\left(\left|x^i\right| - \beta^i\right)^2, & \beta^i < \left|x^i\right| \le \beta^i + \alpha^i \\ \left(\left|x^i\right| - \beta^i - \frac{1}{2}\alpha^i\right), & \left|x^i\right| > \beta^i + \alpha^i \end{cases}$$

with α^i, β^i chosen randomly between $(0, 0.01)$ and $(2, 2.5)$, respectively. Cost function (14) is smooth and convex, and the optimization problem (14) has infinite number of minimizers that correspond to the minimum cost of $f^\star = 0$.

Figure 5 shows the trajectory of decision variables following D-TPD dynamics (7). The error of constraint is shown in Fig. 6. Notice that the decision variable x^i of each agent converges to a point satisfying the constraint only for D-TPD, PD-AL and dac+L∂ algorithms. The error of function value in Fig. 7 shows relatively slow decay for primal-dual based algorithms. We conclude that the decision variables of D-TPD dynamics converges significantly faster to one of the solutions of the optimization problem.

VI. Conclusions

This paper proposed a novel distributed algorithm to solve an in-network optimal resource allocation problem over

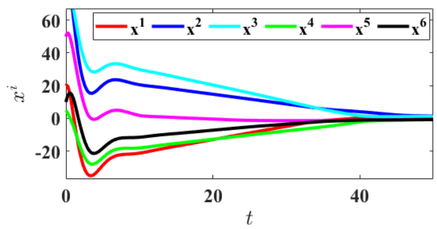

Fig. 5: Trajectories of the D-TPD dynamics for the convex optimization problem (14).

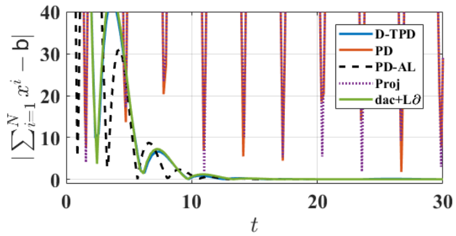

Fig. 6: Error of constraint computed by D-TPD dynamics (blue) and PD (orange), PD-AL(black), Proj (purple), dac+L∂ (green).

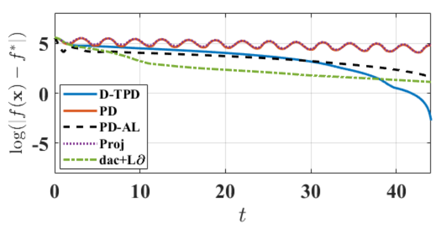

Fig. 7: function error computed by D-TPD dynamics (blue) and PD (orange), PD-AL(black), Proj (purple), dac+L∂ (green).

undirected connected graphs. This algorithm was inspired by a recently proposed first-order centralized algorithm referred to as transformed primal-dual algorithm, which comes with an exponential convergence for a strongly convex cost and asymptotic convergence for a convex cost. We used a control theoretic framework to study the convergence of this algorithm and established that when the local cost functions are strongly convex and have only locally Lipschitz gradients, our proposed distributed algorithm guarantees exponential convergence. For convex local cost functions, the convergence guarantee was asymptotic convergence to a point in the minimizer set. Numerical examples show that our algorithm achieves faster convergence than existing continuous-time first-order distributed solutions for our problem of interest. Our future work focuses on characterizing the stepsize for discrete-time implementation of our algorithm and expanding the results to include multiple demand equations.

References

[1] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[2] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

[3] B. Shi, S. S. Du, M. I. Jordan, and W. J. Su, "Understanding the acceleration phenomenon via high-resolution differential equations," *Mathematical Programming*, 2021. Available online.

[4] L. Chen and J. Wei, "Transformed primal-dual methods for nonlinear saddle point systems," *Journal of Numerical Mathematics*, 2023.

[5] A. J. Wood, F. Wollenberg, and G. B. Sheble, *Power Generation, Operation and Control*. New York: John Wiley, 3rd ed., 2013.

[6] A. Cherukuri and J. Cortés, "Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment," *Automatica*, vol. 74, pp. 183–193, 2016.

[7] J. Chen and V. K. N. Lau, "Convergence analysis of saddle point problems in time varying wireless systems – control theoretical approach," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 443–452, 2012.

[8] A. Ferragut and F. Paganini, "Network resource allocation for users with multiple connections: fairness and stability," *IEEE/ACM Transactions on Networking*, vol. 22, no. 2, pp. 349–362, 2014.

[9] L. Xiao, M. Johansson, and S. P. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Transactions on Communications*, vol. 52, no. 7, pp. 1136–1144, 2004.

[10] R. Madan and S. Lall, "Distributed algorithms for maximum lifetime routing in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 8, pp. 2185–2193, 2006.

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

[12] S. Kar and G. Hug, "Distributed robust economic dispatch in power systems: A consensus + innovations approach," in *Power & Energy Society General Meeting*, (San Diego, CA), pp. 1–8, July 2012.

[13] A. D. Dominguez-Garcia, S. T. Cady, and C. N. Hadjicostis, "Decentralized optimal dispatch of distributed energy resources," in *IEEE Int. Conf. on Decision and Control*, (Hawaii, USA), pp. 3688–3693, Dec. 2012.

[14] A. Esteki and S. S. Kia, "PrivOpt: an intrinsically private distributed optimization algorithm," in *American Control Conference*, (Atlanta, GA, USA), pp. 4262–4267, 2022.

[15] L. Xiao and S. Boyd, "Optimal scaling of a gradient method for distributed resource allocation," *Journal of optimization theory and applications*, vol. 129, no. 3, pp. 469–488, 2006.

[16] Y. Zhang and M. M. Zavlanos, "A consensus-based distributed augmented Lagrangian method," in *IEEE Int. Conf. on Decision and Control*, (CA, USA), 2018.

[17] S. A. Alghunaim, K. Yuan, and A. H. Sayed, "Dual coupled diffusion for distributed optimization with affine constraints," in *IEEE Int. Conf. on Decision and Control*, (FL, USA), 2018.

[18] A. Cherukuri and J. Cortés, "Distributed generator coordination for initialization and anytime optimization in economic dispatch," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 3, pp. 226–237, 2015.

[19] A. Cherukuri and J. Cortés, "Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment," *Automatica*, vol. 74, pp. 183–193, 2016.

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

[21] S. Kia, "Distributed optimal in-network resource allocation algorithm design via a control theoretic approach," *Systems & Control Letters*, vol. 107, pp. 49–57, 2017.

[22] D. Ding and M. Jovanović, "A primal-dual Laplacian gradient flow dynamics for distributed resource allocation problems," in *American Control Conference*, (WI, USA), 2018.

[23] H. Moradian and S. S. Kia, "Cluster-based distributed augmented Lagrangian algorithm for a class of constrained convex optimization problems," *Automatica*, vol. 129, p. 109608, 2021.

[24] K. J. Arrow, L. Hurwicz, and H. Uzawa, *Studies in linear and nonlinear programming*. Stanford University Press, 1958.

[25] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[27] H. Moradian and S. S. Kia, "Cluster-based distributed augmented Lagrangian algorithm for a class of constrained convex optimization problems," *Available on arXiv:1908.06634*, 2020.

[28] W. Haddad and V. Chellaboina, *Nonlinear Dynamical Systems and Control*. Princeton University Press, Princeton, NJ, 2008.

[29] H. K. Khalil, *Nonlinear Systems*. Englewood Cliffs, NJ: Prentice Hall, 3 ed., 2002.

[30] H. Moradian and S. S. Kia, "A distributed continuous-time modified Newton–Raphson algorithm," *Automatica*, vol. 136, p. 109886, 2022.