Distributed Resource Allocation for Multi-Agent Networks

Ola Shorinwa and Mac Schwager

***Abstract*— We present a distributed algorithm for resource allocation problems where each agent computes its optimal resource allocation locally without knowing the resource allocation, objective, and constraints of other agents, guaranteeing the privacy of each agent's local data and resource allocation. Each agent communicates with its neighbors over a point-to-point communication network to satisfy the coupling constraints on the resource allocations of all agents. Our distributed algorithm, derived from the dual formulation of the problem using the consensus alternating direction method of multipliers, applies to resource allocation problems with both convex equality and inequality coupling constraints. As such, unlike many other distributed resource allocation methods, our distributed algorithm is not limited to problems with affine coupling constraints. In addition, our algorithm does not require a feasible initialization of the resource allocations for convergence to an optimal resource allocation. We demonstrate faster empirical convergence of our distributed algorithm to the optimal resource allocation compared to other distributed resource allocation algorithms, with our algorithm converging in about two orders of magnitudes fewer communication rounds.**

***Index Terms*— Distributed Resource Allocation, Distributed Optimization, Resource Sharing, Multi-Agent Systems.**

I. Introduction

RESOURCE allocation problems arise in a variety of applications where groups of agents share a given resource pool to perform their specified tasks, with examples in collaborative target tracking and multi-agent delivery networks. Other applications arise in wireless communications [1]–[4] and economic dispatch of power systems [5]–[7]. Given the limits on the availability of each resource, efficient utilization of the available resources requires coordination among the agents. Such coordination often occurs at a central station which collects all relevant problem information from each agent to compute an optimal allocation of the finite resources. However, these methods suffer from poor scalability to larger groups of agents, considering the inherent communication and computation overhead required by these methods. Further, centralized methods lack robustness to failures as reliance on a central station predisposes the network of agents to a single point of failure at the central station.

Distributed resource allocation methods circumvent these challenges by enabling each agent to compute its optimal resource allocation locally, rather than at a central station. Each agent does not share its local resource allocations with other agents, particularly valuable in competitive scenarios where sharing this information can prove detrimental to an agent. However, many existing distributed methods require each agent to have some global information on the local objective functions of all agents, such as the gradient Lipschitz constant of the objective function, and in some cases on the coupling constraints involving the resource allocations of all agents [8]–[11]. In many situations, obtaining this information in a distributed manner proves challenging, hindering the application of these methods.

A. Contributions

Noting some of the aforementioned limitations of existing distributed methods for resource allocation, we derive a distributed algorithm which addresses these limitations. Our contributions are as follows:

- We provide an algorithm for distributed resource allocation for multi-agent systems (D-RAMS), enabling each agent to compute an optimal resource allocation in convex problems, which satisfies its local constraints, as well as the coupling constraint on the allocations of all agents, without sharing its local data including its local objective, constraints, and resource allocation. While communicating its local dual variables with other neighboring agents over a point-to-point communication network, each agent computes an optimal dual solution, in addition to an optimal resource allocation. Notably, our algorithm does not require knowledge of any form of global information by each agent for its implementation.
- Moreover, under typical conditions of strong convexity of the local objective functions with Lipschitz continuous gradients, our method attains linear convergence of the dual variables to the optimal dual solution, in addition to linear convergence of the resource allocations of all agents to the optimal resource allocation.
- Further, we demonstrate faster empirical convergence of our algorithm compared to a number of other distributed algorithms for the resource allocation problem. Specifically, our algorithm requires about two orders of magnitude fewer

This work was supported by NSF NRI awards 1830402 and 1925030 and ONR award N00014-18-1-2830.*(Corresponding author: Ola Shorinwa).*

O. Shorinwa is with the Department of Mechanical Engineering, Stanford University, Stanford, CA, USA, (email: shorinwa@stanford.edu).

M. Schwager is with the Department of Aeronautics and Astronautics Engineering, Stanford University, Stanford, CA, USA, (email: schwager@stanford.edu).

communication rounds to converge to the optimal resource allocation compared to the best competing distributed resource allocation algorithms [12]–[14] in our simulation studies, with a convergence threshold of $1e^{-4}$.

II. Related Work

Resource allocation problems have been solved through center-free methods, a class of distributed subgradient methods where neighboring agents exchange their local subgradients to update their local resource allocations [15]–[17]. In general, these methods exhibit $O(1/k)$ convergence rates, with linear convergence for strongly convex objective functions that are Lipschitz smooth. However, these methods require the initial resource allocations of all agents to satisfy the coupling constraint on the agents' allocations, posing a significant challenge.

Dual decomposition methods formulate the optimization problem in its dual form and compute the optimal dual solutions from which the optimal resource allocations can be obtained assuming strong duality holds [22]–[25]. However, updates to the dual variables through dual ascent depend on the resource allocations of all agents, and thus require a central station [26]. Distributed dual decomposition methods enable each agent to compute the dual variables locally and enforce agreement among all the local dual variables through a linear consensus [12], [19] or push-sum consensus [27] scheme, with a convergence rate of $O(\ln(k)/\sqrt{k})$ to the optimal resource allocation where k represents the number of iterations. For a faster $O(1/k)$ convergence rate, some methods apply gradient-tracking schemes which involve linear consensus on the subgradients [8], [21], while others utilize Nesterov accelerated gradient ascent in the dual update [20].

The consensus alternating direction method of multipliers (C-ADMM) [28] has been applied to the dual formulation of resource allocation problems in [29] with an $O(1/k)$ convergence rate. In this method, each agent relaxes its local affine inequality constraints and maintains additional slack and dual variables for these local constraints. Considering that the distributed algorithm in [29] applies only to problems with affine equality coupling constraints, the dual-based C-ADMM algorithm in [30] extends the method in [29] to problems where the affine coupling constraint lies in a nonempty, closed, and convex cone, including problems with affine inequality coupling constraints. However, both algorithms remain limited to resource allocation problems with affine coupling constraints. In addition, the authors of [30] do not provide the convergence rate of their algorithm. The Tracking-ADMM algorithm in [31] blends ADMM with gradient tracking schemes to allow for fully distributed update procedures. However, like [29], [30], the Tracking-ADMM algorithm [31] is not amenable to problems with inequality constraints. Under the assumption of strongly convex and smooth objective functions, the distributed algorithms based on gradient tracking [8], [21] provide a linear convergence rate.

Distributed primal-dual methods iteratively update each agent's resource allocation through gradient descent and its dual variables through gradient ascent [9], [32], [33], with a subsequent orthogonal projection of the resource allocations and dual variables to satisfy local constraints. Like dual decomposition methods, primal-dual approaches require strong duality for convergence to a saddle-point of the problem. In some primal-dual methods, each agent maintains a copy of the resource allocations of all agents [10], creating significant computation and communication challenges as the number of agents involved in the problem grows. Other primal-dual methods provide better scaling performance by enabling each agent to only compute its resource allocation and dual variables with a convergence rate of $O(1/\sqrt{k})$ [13]. By utilizing accelerated gradient descent schemes, some primal-dual methods achieve a faster convergence rate of $O(1/k)$ [34]. In [14], [35], each agent updates its local variables using a primal-dual algorithm based on [36] which bears some relations to proximal-point methods such as the alternating direction method of multipliers, with an $O(1/k)$ convergence rate. The primal-dual method in [18] introduces surplus variables to allow for the satisfaction of the coupling constraint between the agents' resource allocations. However, this method requires a feasible initialization with respect to the coupling constraint, posing a challenge. Other primal-dual methods introduce perturbations in the update procedures where each agent updates its primal and dual variables using perturbed subgradients [11], with less stringent assumptions required for convergence to the saddle points of the problem.

We derive our method by applying C-ADMM to the dual problem, under the typical condition that strong duality holds. While some distributed methods require knowledge of the Lipschitz constant of the objective function [8], [9], [17], [20] and others require knowledge of a common vector satisfying Slater's condition by each agent [10], [11] which poses challenges for networks of distributed agents, our method does not require knowledge of any global information. Further, our method applies to resource allocation problems with any set of local convex constraints, while many distributed methods only consider problems with no local constraints [9], [17] and problems with local box constraints [18]. Moreover, our method applies to both equality and inequality coupling constraints on the resource allocations, unlike many distributed methods which apply only to problems with an equality coupling constraint [8], [18], [19].

Table I summarizes the convergence rates and requirements imposed by a number of notable distributed algorithms for resource allocation problems, as well as the classes of problems amenable to these algorithms, in comparison to our distributed algorithm. The paper proceeds as follows: We introduce the resource allocation problem in Section IV including local constraints for each agent. In Section V, we derive a distributed method for the resource allocation problem and prove convergence to the optimal solution. We demonstrate our method in resource allocation problems where the resource allocation of each agent must satisfy its local constraints in addition to problems without local constraints, comparing its performance to those of other distributed methods in Section IX. We conclude in Section X.

TABLE I

A SUMMARY OF THE REQUIREMENTS, CONVERGENCE RATES, AND CLASSES OF PROBLEMS AMENABLE TO SOME NOTABLE DISTRIBUTED ALGORITHMS FOR RESOURCE ALLOCATION PROBLEMS IN COMPARISON TO OUR DISTRIBUTED ALGORITHM. THE BOOLEAN ENTRIES INDICATE IF A SPECIFIC ALGORITHM IS AMENABLE TO PROBLEMS WITH A LOCAL CONSTRAINT OR REQUIRES KNOWLEDGE OF THE LIPSCHITZ CONSTANT OF THE OBJECTIVE FUNCTION AT INITIALIZATION OF THE ALGORITHM. WE DENOTE A MISSING ENTRY BY "-" WHEN THE REQUIRED INFORMATION IS NOT PROVIDED BY THE AUTHORS OF THE ALGORITHM IN THEIR PAPER.

Algorithm	Coupling Constraint	Local Constraint	Convergence Rate	Lipschitz Constant Req.
[18]	Equality	✓(Box)	-	✗
[8]	Equality	✓	Linear	✓
[19]	Equality	✓	Sub-linear, $O(\ln(k)/\sqrt{k})$	✗
[17]	Equality	✗	Linear	✓
[20]	(In)Equality	✓	Sub-linear, $O(1/k)$	✓
[14]	(In)Equality	✓	Sub-linear, $O(1/k)$	✓
[21]	Equality	✓	Linear	✓
[12]	(In)Equality	✓	Sub-linear	✗
Ours	(In)Equality	✓	Linear	✗

III. Notation and Preliminaries

We denote the weighted norm $x^\mathsf{T} W x$ as $\|\cdot\|_W^2$ with $W > 0$ and represent the cardinality of a set $\mathcal{D}$ as $|\mathcal{D}|$. Likewise, $\sigma_{\max}(A)$ denotes the maximum non-zero singular value of matrix A, with $\sigma_{\min}(A)$ denoting its minimum non-zero singular value. We denote the domain of a function f as $\mathrm{dom}(f)$ and the identity matrix as $I_n \in \mathbb{R}^{n\times n}$. We denote the subdifferential of a scalar-valued function $f: \mathbb{R}^n \to \mathbb{R}$ at $x \in \mathbb{R}^n$ as $\partial f(x)$. When f is differentiable, we denote the gradient of f at x by $\nabla f(x)$, and the Jacobian of a vector-valued function f at x as $J_f(x)$.

Definition 1 (Q-Linear Convergence)**.** *Given that a sequence $\{x^k\}$ converges to a stationary point $x^\star$, we describe linear convergence of $\{x^k\}$ as Quotient-linear (Q-linear) if there exists $\beta \in (0,1)$ with*

$$\frac{\|x^{k+1} - x^\star\|}{\|x^k - x^\star\|} \leq \beta.$$

Definition 2 (R-Linear Convergence)**.** *We describe linear convergence of the sequence $\{x^k\}$ to a stationary point $x^\star$ as Root-linear (R-linear) if*

$$\|x^{k+1} - x^\star\| \leq p^k, \tag{1}$$

and $\{p^k\}$ converges to zero Q-linearly.

Definition 3 (Closed Function)**.** *A function f is closed if, for each $\alpha \in \mathbb{R}$, the set $\{x \in \mathrm{dom}(f) \mid f(x) \leq \alpha\}$ is closed.*

Definition 4 (Strongly Convex Function)**.** *A subdifferentiable function $f: \mathbb{R}^n \to \mathbb{R}$ is λ-strongly convex if there exists a constant $\lambda > 0$ such that*

$$f(y) \geq f(x) + h^\mathsf{T}(y-x) + \lambda \|y - x\|_2^2 \tag{2}$$

for all x, $y \in \mathrm{dom}(f)$ and $h \in \partial f(x)$ is a subgradient of f at x.

Definition 5 (Lipschitz Continuous Function)**.** *A function $f: \mathbb{R}^n \to \mathbb{R}^m$ is a λ-Lipschitz continuous function if there exists a constant λ such that*

$$\|f(x) - f(y)\|_2^2 \leq \lambda \|x - y\|_2^2 \tag{3}$$

for all x, $y \in \mathrm{dom}(f)$.

Definition 6 (Smooth Function)**.** *A function $f: \mathbb{R}^n \to \mathbb{R}^m$ is λ-Lipschitz smooth if f is continuously differentiable and has Lipschitz continuous gradients, i.e., there exists a constant $\lambda > 0$ such that*

$$\|\nabla f(x) - \nabla f(y)\|_2^2 \leq \lambda \|x - y\|_2^2 \tag{4}$$

for all x, $y \in \mathrm{dom}(f)$.

Communication Network

We represent the communication network between the agents as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a set of vertices $\mathcal{V} = \{1, \cdots, N\}$, consisting of N agents, and a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, where the edges in $\mathcal{E}$ denote a communication link between a pair of agents. We denote the neighbor set of agent i, which consists of agents which share a communication link with agent i, as $\mathcal{N}_i$.

IV. Problem Formulation

We consider a resource allocation problem among a group of N agents, given by

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \sum_{i=1}^{N} f_i(x_i) \\ \text{subject to} \quad & \sum_{i=1}^{N} g_i(x_i) \leq 0 \\ & h_i(x_i) = 0 \quad \forall i \in \mathcal{V} \\ & r_i(x_i) \leq 0 \quad \forall i \in \mathcal{V} \end{aligned} \tag{5}$$

where $x_i \in \mathbb{R}^{n_i}$ denotes the resource allocation of agent i, $f_i: \mathbb{R}^{n_i} \to \mathbb{R}$ represents the objective function of agent i, $g_i: \mathbb{R}^{n_i} \to \mathbb{R}^m$ represents the constraint function coupling the resource allocation of agent i to the allocations of all the agents, and $\boldsymbol{x} = [x_i^\mathsf{T}, \forall i \in \mathcal{V}]^\mathsf{T}$ denotes the concatenation of the resource allocations of all agents. In addition, the resource allocation of agent i must satisfy its local constraints specified by $h_i: \mathbb{R}^{n_i} \to \mathbb{R}^{a_i}$ and $r_i: \mathbb{R}^{n_i} \to \mathbb{R}^{b_i}$. We denote the feasible set of agent i's resource allocation as $\mathcal{X}_i = \{x_i \mid h_i(x_i) = 0, r_i(x_i) \leq 0\}$, with $\boldsymbol{\mathcal{X}} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_N$ denoting the feasible set of the resource allocations of all agents. We denote the composite objective function and composite coupling constraint function as

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} f_i(x_i), \quad g(\boldsymbol{x}) = \sum_{i=1}^{N} g_i(x_i), \tag{6}$$

respectively. In this work, we assume that the objective and coupling constraint functions are convex, with the additional assumption that $\mathcal{X}_i$ represents a convex set, $\forall i \in \mathcal{V}$. These assumptions guarantee convexity of the resource allocation problem in (5). Note that this formulation encompasses most resource allocation problems, including problems with equality coupling constraints. Problems with equality coupling constraints can be readily transformed into the same form as (5) by replacing the equality constraint with the pair of inequalities $\geq$ and $\leq$.

V. Distributed Resource Allocation

Noting the significant computational and communication challenges faced by centralized methods in solving (5), we derive a distributed method for the resource allocation problem. We assume subdifferentiability of the objective and coupling constraint functions.

Assumption 1. *The objective function f_i and coupling constraint function g_i are subdifferentiable convex functions, $\forall i \in \mathcal{V}$.*

From Assumption 1, the subdifferentials of the objective and coupling constraints functions are non-empty. Note that we do not require differentiability of the objective and coupling constraint functions. In addition, we assume that Slater's condition applies to the problem in (5).

Assumption 2 (Slater's Condition)**.** *A feasible resource allocation $\tilde{\boldsymbol{x}}$ exists in the relative interior of $\boldsymbol{\mathcal{X}} \cap \mathrm{dom}(f)$ and satisfies $\sum_{i=1}^{N} g_i(\tilde{x}_i) \leq 0$ for affine components of $g_i(\cdot)$ and $\sum_{i=1}^{N} g_i(\tilde{x}_i) < 0$ for all other components.*

The existence of a resource allocation satisfying Slater's condition indicates that strong duality holds, i.e., a saddle-point exists, and the set of dual solutions is non-empty. Moreover, we assume that the dual optimal value is attained.

We obtain the Lagrangian of the resource allocation problem (5) given by

$$\mathcal{L}(\boldsymbol{x}, y) = \sum_{i=1}^{N} f_i(x_i) + \sum_{i=1}^{N} y^{\mathsf{T}} g_i(x_i) \tag{7}$$

where $y \in \mathbb{R}^m$ denotes the dual variable for the coupling constraint on the resource allocations of all agents with $y \geq 0$ and $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$. The dual function for the Lagrangian is given by

$$v(y) = \inf_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, y) \tag{8}$$

which simplifies to

$$v(y) = \inf_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}} \sum_{i=1}^{N} \left(f_i(x_i) + y^{\mathsf{T}} g_i(x_i) \right) \tag{9}$$

for the Lagrangian (7). From separability of $\boldsymbol{x}$, the dual function can be expressed as

$$v(y) = \sum_{i=1}^{N} \inf_{x_i \in \mathcal{X}_i} \left\{ f_i(x_i) + y^{\mathsf{T}} g_i(x_i) \right\} \tag{10}$$

which highlights the separable structure of the dual function, with coupling arising from the common dual variable. In the special case with affine coupling constraints given by

$$g_i(x_i) = C_i x_i - d_i, \tag{11}$$

where $C_i \in \mathbb{R}^{m \times n_i}$ and $d \in \mathbb{R}^m$, we can express the dual function in (10) as

$$v(y) = \sum_{i=1}^{N} \left(-f_i^{\star}(-C_i^{\mathsf{T}} y) - y^{\mathsf{T}} d_i \right) \tag{12}$$

where $f_i^{\star}$ denotes the Fenchel conjugate of f_i, with

$$f_i^{\star}(y) = \sup_{x_i \in \mathcal{X}_i} \left\{ y^{\mathsf{T}} x_i - f_i(x_i) \right\}. \tag{13}$$

The dual problem consists of maximizing the dual function, described by

$$\underset{y \in \mathcal{Y}}{\text{maximize}} \ \sum_{i=1}^{N} \phi_i(y) \tag{14}$$

where $\mathcal{Y}$ denotes the feasible set of the dual problem, i.e., $\mathcal{Y} = \{y \mid y \geq 0\}$, and $\phi_i(y) = \inf_{x_i \in \mathcal{X}_i} \left\{ f_i(x_i) + y^{\mathsf{T}} g_i(x_i) \right\}$.

Given the existence of a saddle-point for the resource allocation problem (5),

$$\mathcal{L}(\boldsymbol{x}^{\star}, y) \leq \mathcal{L}(\boldsymbol{x}^{\star}, y^{\star}) \leq \mathcal{L}(\boldsymbol{x}, y^{\star}) \tag{15}$$

for any feasible pair of resource allocation $\boldsymbol{x}$ and dual variable y where

$$\boldsymbol{x}^{\star} \in \underset{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}}{\mathrm{arginf}} \ \mathcal{L}(\boldsymbol{x}, y^{\star}) \quad \text{and} \quad y^{\star} \in \underset{y \in \mathcal{Y}}{\mathrm{argsup}} \ \mathcal{L}(\boldsymbol{x}^{\star}, y). \tag{16}$$

Since strong duality holds, we can solve the dual optimization problem (14) to obtain an optimal dual solution and subsequently recover an optimal primal solution for the resource allocation problem in (5). However, the dual problem (14) involves the global dual variable y, which is unavailable to any individual agent. To derive a distributed method for computing the dual variable, we assign a local copy of the dual variable to each agent with a local equality constraint between neighboring agents to ensure that all agents compute the same dual variables. In addition, we express the dual function in terms of these local copies with

$$v(\boldsymbol{y}) = \sum_{i=1}^{N} \phi_i(y_i). \tag{17}$$

With this approach, the agents compute their dual variables from the resulting problem described by

$$\begin{aligned} &\underset{\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}}{\text{maximize}} \ \sum_{i=1}^{N} \phi_i(y_i) \\ &\text{subject to } y_i = y_j \quad \forall (i,j) \in \mathcal{E} \end{aligned} \tag{18}$$

where agent i computes only $y_i \in \mathbb{R}^m$ and $\boldsymbol{y} = [y_i^{\mathsf{T}}, \forall i \in \mathcal{V}]^{\mathsf{T}}$ denotes the concatenation of the local dual variables of all agents. Note that the equality constraint between the local dual variables only exists between neighboring agents, represented by the edges in $\mathcal{E}$.

Proposition 1. *The dual problem in* (18) *is equivalent to the original problem in* (14) *with the same optimal solution and optimal objective value.*

Proof. Since the communication graph $\mathcal{G}$ is connected, each agent shares an equality constraint on its local dual variable with at least one other agent. At any feasible $\boldsymbol{y}$ in (18), all the agents have the same local dual variables, with the objective function in (18) simplifying to the same objective function in (14). As a result, since the optimization problems in (18) and (14) have the same feasible set, both problems have the same optimal solution with the same optimal objective value. □

While typical dual decomposition and primal-dual methods update the dual variables through gradient ascent, we derive update procedures for the dual variables using C-ADMM [28]. We introduce local slack variables α and γ into the equality constraints in (18), resulting in the problem

$$\begin{aligned} \underset{\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}}{\text{maximize}} \ & \sum_{i=1}^{N} \phi_i(y_i) \\ \text{subject to } & y_i = \alpha_{ij} \quad \forall (i,j) \in \mathcal{E} \\ & y_j = \gamma_{ij} \quad \forall (i,j) \in \mathcal{E} \\ & \alpha_{ij} = \gamma_{ij} \quad \forall (i,j) \in \mathcal{E} \end{aligned} \tag{19}$$

where $\alpha_{ij} \in \mathbb{R}^m$ denotes the local slack variable of agent i and $\gamma_{ij} \in \mathbb{R}^m$ denotes the local slack variable of agent j for the equality constraint between the pair of neighboring agents i and j to allow for separability of the augmented Lagrangian of the dual problem in (19), which is given by

$$\begin{aligned} \mathcal{L}_a(\boldsymbol{y}, \alpha, \gamma, u, w) = & \sum_{i=1}^{N} \phi_i(y_i) \\ & - \sum_{(i,j) \in \mathcal{E}} \left(u_{ij}^{\mathsf{T}} \left(y_i - \alpha_{ij} \right) + w_{ij}^{\mathsf{T}} \left(y_j - \gamma_{ij} \right) \right) \\ & - \frac{\rho}{2} \sum_{(i,j) \in \mathcal{E}} \left(\left\| y_i - \alpha_{ij} \right\|_2^2 + \left\| y_j - \gamma_{ij} \right\|_2^2 \right) \end{aligned} \tag{20}$$

where $u_{ij} \in \mathbb{R}^m$ and $w_{ij} \in \mathbb{R}^m$ denote Lagrange multipliers for the equality constraints on y_i and y_j in (19). In addition, we do not relax the equality constraints between α and γ and the constraint $\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}$. Rather, we enforce these constraints in deriving the update procedures for these variables. The augmented Lagrangian includes the extra quadratic terms on the violations of the constraints between corresponding slack variables, with ρ defining a penalty weighting the contributions of these violations to the augmented Lagrangian. When all agents compute the same values of y, the contribution of these quadratic terms vanishes.

In our distributed algorithm, the agents update $\boldsymbol{y}$ and the slack variables as the maximizers of the augmented Lagrangian using the Lagrange multipliers at the previous iteration, before updating the Lagrange multipliers through gradient descent, in an alternating manner, where the update to $\boldsymbol{y}$ precedes the update of the slack variables. With this update scheme, agent i updates its local dual variable y_i using

$$\begin{aligned} y_i^{k+1} \in \underset{y_i \in \mathcal{Y}}{\text{argmax}} \Big\{ & \phi_i(y_i) - \sum_{j \in \mathcal{N}_i} \left(u_{ij}^k + w_{ji}^k \right)^{\mathsf{T}} y_i \\ & - \frac{\rho}{2} \sum_{j \in \mathcal{N}_i} \left(\left\| y_i - \alpha_{ij}^k \right\|_2^2 + \left\| y_i - \gamma_{ji}^k \right\|_2^2 \right) \Big\} \end{aligned} \tag{21}$$

at iteration k. The update procedure for the slack variables simplifies to

$$\alpha_{ij}^{k+1} = \gamma_{ij}^{k+1} = \frac{1}{2} \left(y_i^{k+1} + y_j^{k+1} \right), \tag{22}$$

when the Lagrange multipliers are initialized with $u_{ij}^0 = w_{ij}^0 = 0, \ \forall (i,j) \in \mathcal{E}$.

Likewise, the update procedures for the Lagrange multipliers simplify to

$$u_{ij}^{k+1} = u_{ij}^k + \frac{\rho}{2} \left(y_i^{k+1} - y_j^{k+1} \right) \tag{23}$$

and

$$w_{ij}^{k+1} = w_{ij}^k + \frac{\rho}{2} \left(y_j^{k+1} - y_i^{k+1} \right) \tag{24}$$

using (22), with $u_{ij}^k = -w_{ij}^k$ at all iterations k. The closed-form solutions for α_{ij}^k and γ_{ij}^k enable us to further simplify the optimization problem in (21), resulting in

$$\begin{aligned} y_i^{k+1} \in \underset{y_i \in \mathcal{Y}}{\text{argmax}} \Bigg\{ & \phi_i(y_i) - q_i^{k\mathsf{T}} y_i \\ & - \rho \sum_{j \in \mathcal{N}_i} \left\| y_i - \frac{y_i^k + y_j^k}{2} \right\|_2^2 \Bigg\} \end{aligned} \tag{25}$$

where

$$q_i^k = \sum_{j \in \mathcal{N}_i} \left(u_{ij}^k + w_{ji}^k \right) \tag{26}$$

combines the Lagrange multipliers u and w, with its associated update procedure given by

$$q_i^{k+1} = q_i^k + \rho \sum_{j \in \mathcal{N}_i} \left(y_i^{k+1} - y_j^{k+1} \right) \tag{27}$$

at iteration k.

With the simplified procedures in (29) and (27), each agent does not compute the slack variables α and γ and the Lagrange multipliers u and w. We provide additional details on the derivations of the update procedures in the Appendix.

The update procedure for the dual variable y_i requires solving the optimization problem

$$\begin{aligned} \underset{y_i \in \mathcal{Y}}{\text{maximize}} \ \underset{x_i \in \mathcal{X}_i}{\text{minimize}} \ \Bigg\{ & f_i(x_i) + y_i^{\mathsf{T}} g_i(x_i) - q_i^{k\mathsf{T}} y_i \\ & - \rho \sum_{j \in \mathcal{N}_i} \left\| y_i - \frac{y_i^k + y_j^k}{2} \right\|_2^2 \Bigg\} \end{aligned} \tag{28}$$

which represents a min-max optimization problem over the variables x_i and y_i. Although min-max optimization problems are difficult to solve generally, we can leverage the existence of a saddle-point to derive a closed-form solution for y_i by swapping the order of the optimization problems in (28). The resulting closed-form solution for y_i at iteration k is given by

$$y_i^{k+1} = \frac{1}{2\rho|\mathcal{N}_i|} \max\big(0,\ \psi_i^k(x_i^{k+1})\big) \tag{29}$$

with

$$\psi_i^k(x_i) = g_i(x_i) - q_i^k + \rho \sum_{j \in \mathcal{N}_i} \big(y_i^k + y_j^k\big) \tag{30}$$

where agent i computes x_i^{k+1} as the solution to the problem

$$\underset{x_i \in \mathcal{X}_i}{\text{minimize}} \left\{ f_i(x_i) + \frac{1}{4\rho|\mathcal{N}_i|} \left\|\max(0, \psi_i^k(x_i))\right\|_2^2 \right\} \tag{31}$$

at iteration k. Note that the minimization problem represents a convex optimization problem, from convexity of f_i and g_i. We provide more details on the derivation of the closed-form update procedures for y_i in the Appendix. In deriving our distributed algorithm, we relax the local equality constraints on the dual variables maintained by each agent. Consequently, the values of these local copies may not be equal at an intermediate iteration, before an optimal solution is computed. However, upon convergence of our algorithm, which we discuss in Section VI, the local copies of the dual variable satisfy the local equality constraints and, thus, have the same values.

We outline our algorithm for distributed resource allocation problems in Algorithm 1.

Algorithm 1: Distributed Resource Allocation for Multi-Agent Systems (D-RAMS)

Initialization:
$x_i^0 \in \mathbb{R}^{n_i}$, $y_i^0 \in \mathbb{R}^m \quad \forall i \in \mathcal{V}$
$q_i^0 \leftarrow 0 \quad \forall i \in \mathcal{V}$
$k \leftarrow 0$
do in parallel $\forall i \in \mathcal{V}$
  $x_i^{k+1} \leftarrow$ Procedure (31)
  $y_i^{k+1} \leftarrow$ Procedure (29)
  $q_i^{k+1} \leftarrow$ Procedure (27)
  $k \leftarrow k + 1$
while *not converged or stopping criterion is not met*;

Communication Complexity

With D-RAMS, each agent shares its local dual variables with its neighbors for the update procedure in (27). As such, each communication round involves sharing $8m$ bytes of information, assuming double-precision floating-point representation of the dual variables, where m denotes the output dimension of the coupling constraint function. Hence, D-RAMS achieves the same communication complexity with dual decomposition methods, improving upon the communication complexity of other distributed methods which utilize consensus on the primal and dual variables.

VI. Convergence Analysis

In this section, we prove convergence of our algorithm for subdifferentiable convex objective and constraint functions.

Theorem 1 (Sub-linear Convergence of $\{\boldsymbol{y}^k\}$)**.** *The dual iterates $\{y_i^k\}$ of agent i converge to an optimal dual solution $y^\star$, $\forall i \in \mathcal{V}$, sub-linearly.*

Proof. We note that the proof of the sub-linear convergence of ADMM is provided in many references [37], [38]. Consequently, we omit the proof here and refer readers to the aforementioned references. □

Note that the convergence results in Theorem 1 does not require differentiability or strong convexity of the objective and constraint functions. We can obtain stronger results on the convergence rates of our algorithm, if we make additional assumptions on the objective and constraint functions. Specifically, if the objective function f_i is closed and strongly convex with Lipschitz continuous gradients, the coupling constraint function g_i is affine, and $\mathcal{X}_i = \mathbb{R}^{n_i}$, $\forall i \in \mathcal{V}$, then our distributed algorithm converges at a linear rate.

Before proceeding, we provide the following lemmas on Fenchel duality. We refer readers to [39], [40] for the proof of these results.

Lemma 1. *If f_i is closed and λ-strongly convex, then its Fenchel conjugate $f_i^\star$ is $\frac{1}{\lambda}$-strongly smooth.*

Lemma 2. *If a convex function f_i has Lipschitz continuous gradients with parameter L, then its Fenchel conjugate $f_i^\star$ is $\frac{1}{L}$-strongly convex.*

With the assumption of an affine coupling constraint function, given in (11), ϕ_i simplifies to the expression given in (12), which involves the Fenchel conjugate $f_i^\star$ of f_i. If the objective function f_i of agent i is $\frac{1}{L_i}$-strongly convex (and closed) with $\frac{1}{\lambda_i}$-Lipschitz continuous gradients, Lemmas 1 and 2 indicate that $f_i^\star$ is λ_i-strongly convex and L_i-strongly smooth. Particularly, the composite dual function is continuously differentiable and λ-strongly convex and has Lipschitz continuous gradients with parameter L, where $\lambda = \min\{\lambda_i, \forall i \in \mathcal{V}\}$ and $L = \max\{L_i, \forall i \in \mathcal{V}\}$.

We state the linear convergence rates of the dual iterates generated by our algorithm in the following theorem.

Theorem 2 (Linear Convergence of $\{\boldsymbol{y}^k\}$)**.** *The local dual iterates of each agent $\{y_i^k\}$, $\forall i \in \mathcal{V}$, converge R-linearly to the optimal dual variable $y^\star$, for a λ-strongly concave dual function with L-Lipschitz continuous gradients.*

Proof. We provide the proof in the Appendix. □

We note that the convergence rate of the dual iterates depends on the connectedness of the underlying communication network. We discuss this relationship in the Appendix. In the following theorem, we show convergence of the sequence of resource allocations computed by each agent to the optimal resource allocation $x^\star$. We keep the same assumptions on differentiability of the local objective function of each agent. Further, in problems with strongly convex objective and coupling constraint functions, we prove linear convergence of the sequence of resource allocations to the optimal resource allocation $x^\star$.

Theorem 3 (Convergence of $\{\boldsymbol{x}^k\}$)**.** *The sequence of resource allocation $\{x_i^k\}$ of agent i converges to an optimal resource allocation $x_i^\star$, $\forall i \in \mathcal{V}$, for a differentiable objective function f_i. Moreover, for a strongly convex objective function and an affine coupling constraint function, the sequence of resource allocation $\{x_i^k\}$ converges R-linearly to the optimal resource allocation $x_i^\star$, $\forall i \in \mathcal{V}$.*

Proof. We provide the proof in the Appendix. □

In addition, we provide the following theorem on the convergence rate of the violation of the coupling constraint in problems with an affine equality coupling constraint.

Theorem 4. *The violation of the coupling constraint on the resource allocation of all agents $\boldsymbol{x}$ converges R-linearly to zero, in problems with an affine equality coupling constraint.*

Proof. We provide the proof in the Appendix. □

We note that our distributed algorithm does not require a feasible initialization of the resource allocation of each agent and its dual variables, as the local variables of each agent converges to the optimal solution irrespective of its initialization. Moreover, each agent does not share its resource allocations with other agents, providing privacy in situations where disclosing this information can be detrimental to the agent. Each agent only shares its local dual variables with its neighbors.

VII. Equality Constrained Resource Allocation Problems

In this section, we consider a subclass of resource allocation problems where the inequality coupling constraint is replaced by an equality constraint, given by

$$\sum_{i=1}^{N} g_i(x_i) = 0. \tag{32}$$

For this subclass of problems, the update procedure for the dual variable y_i of agent i in (25) reduces to an unconstrained optimization problem in y_i. As a result, agent i computes its local dual variable from the problem

$$\begin{aligned} y_i^{k+1} \in \operatorname*{argmax}_{y_i} \Bigg\{ & \phi_i(y_i) - q_i^{k\mathsf{T}} y_i \\ & - \rho \sum_{j \in \mathcal{N}_i} \left\| y_i - \frac{y_i^k + y_j^k}{2} \right\|_2^2 \Bigg\} \end{aligned} \tag{33}$$

which admits a closed-form solution for y_i at iteration k, given by

$$y_i^{k+1} = \frac{1}{2\rho|\mathcal{N}_i|} \psi_i\left(x_i^{k+1}\right), \tag{34}$$

following the same approach utilized in Section V, with $\psi_i(\cdot)$ defined in (30). Using (34) in (28), we obtain a simpler update procedure for the resource allocation of agent i at iteration k, with the associated optimization problem given by

$$\operatorname*{minimize}_{x_i \in \mathcal{X}_i} \left\{ f_i(x_i) + \frac{1}{4\rho|\mathcal{N}_i|} \left\| \psi_i^k(x_i) \right\|_2^2 \right\}, \tag{35}$$

which represents a convex optimization problem, noting that the coupling constraint function g_i must be affine in convex resource allocation problems. Subsequently, agent i updates its Lagrange multipliers using (27).

In equality constrained resource allocation problems with quadratic or linear objective functions where $\mathcal{X}_i = \mathbb{R}^n$, the update procedure for x_i in (35) admits a closed-form solution. As a result, each agent can update its local variables efficiently, without resorting to iterative optimization methods such as interior-point methods, cutting-plane methods, or bundle methods to compute its resource allocation. For other problems, we discuss approaches to achieve lower computation complexity, in the next section.

VIII. D-RAMS: Approximate Updates

Although each agent can compute its dual variables in closed-form, updating its primal variables might require significant computational effort, depending on its local objective function in (31). To reduce the computational complexity of the update procedures, we derive variants of our distributed algorithm D-RAMS with a simpler update procedure for the primal variable, enabling each agent to solve its local optimization problem efficiently. We replace the optimization problem in (31) with its approximation, given by

$$\begin{aligned} x_i^{k+1} = \operatorname*{argmin}_{x_i \in \mathcal{X}_i} \Big\{ & \mathcal{J}_i\left(x_i^k\right) + \nabla \mathcal{J}_i\left(x_i^k\right)^\mathsf{T} \left(x_i - x_i^k\right) \\ & + \frac{1}{2}\left(x_i - x_i^k\right)^\mathsf{T} \mathcal{H}_i\left(x_i^k\right)\left(x_i - x_i^k\right) \Big\} \end{aligned} \tag{36}$$

where

$$\mathcal{J}_i(x_i) = f_i(x_i) + \frac{1}{4\rho|\mathcal{N}_i|} \left\| \max(0, \psi_i^k(x_i)) \right\|_2^2 \tag{37}$$

and

$$\nabla \mathcal{J}_i(x_i) = \nabla f_i(x_i) + \frac{1}{2\rho|\mathcal{N}_i|} \left(J_{g_i}(x_i)\right)^\mathsf{T} \max(0, \psi_i^k(x_i)). \tag{38}$$

We denote the Hessian of $\mathcal{J}_i$ evaluated at x_i or its positive definite approximation as $\mathcal{H}_i(x_i) \in \mathbb{R}^{n_i \times n_i}$. Agent i can solve the optimization problem in (36) only once to compute an approximate solution for x_i^{k+1}, before updating its dual variables and Lagrange multipliers. However, repeating the approximation scheme multiple times, where the accuracy of each successive approximation improves upon that of the preceding approximation, could improve the convergence rate of our algorithm by producing more accurate iterates during the execution of D-RAMS. We discuss possible choices of $\mathcal{H}_i(x_i)$ in (36).

A. Linearized Updates

To minimize the computational complexity of (31), we can define $\mathcal{H}_i(x_i)$ as

$$\mathcal{H}_i(x_i) = \frac{1}{\beta_i} I_{n_i} \tag{39}$$

where $I_{n_i} \in \mathbb{R}^{n_i \times n_i}$ and $\beta_i \in \mathbb{R}$, with $\beta_i > 0$, resulting in an approximation for $\mathcal{H}_i(x_i)$ independent of the local primal variable of agent i. By eschewing the computation of the exact Hessian of $\mathcal{J}_i(x_i)$, this approximation scheme only requires each agent to compute the gradient of its local objective function in updating its primal variables. The resulting optimization problem can be solved efficiently using proximal gradient descent, particularly when the proximal operator associated with the feasible set can be evaluated easily, or when $\mathcal{X}_i = \mathbb{R}^{n_i}$. As a result, we can interpret β_i as the step-size in the proximal gradient descent update.

B. Quadratic Updates

To obtain a more accurate approximation of the optimization problem in (31), we can compute the exact Hessian of $\mathcal{J}_i(x_i)$ for $\mathcal{H}_i(x_i)$, given by

$$\begin{aligned} \mathcal{H}_i(x_i) &= \nabla^2 f_i(x_i) \\ &\quad + \frac{1}{2\rho|\mathcal{N}_i|} \sum_{r=1}^{m} \eta_r \Big(\nabla \psi_{i,r}^k(x_i) \big(\nabla \psi_{i,r}^k(x_i)\big)^\mathsf{T} \\ &\qquad + \nabla \psi_{i,r}^k(x_i) \nabla^2 \psi_{i,r}^k(x_i) \Big) \end{aligned} \tag{40}$$

where $\psi_{i,r}^k$ denotes the r-th component of the coupling constraint function of agent i and

$$\eta_r = \begin{cases} 1 & \text{if } \psi_{i,r}^k(x_i) > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{41}$$

Alternatively, we can utilize a finite difference approximation scheme for the Hessian, which generally involve approximating the Hessian from the finite difference of the local gradient and primal variables of each agent (such as the Broyden-Fletcher-Goldfarb-Shanno (BFGS) and Davidon-Fletcher-Powell (DFP) schemes [41]). Although this approach requires greater computational effort, the updates can result in improved convergence rates. In addition, this approximation scheme enables us to derive closed-form solutions for x_i, when the problem consists of local equality constraints, or when $\mathcal{X}_i = \mathbb{R}^{n_i}$.

Algorithm 2 outlines the update procedures performed by each agent in computing its resource allocation.

Algorithm 2: Distributed Resource Allocation for Multi-Agent Systems via Approximate Updates (D-RAMS-Ax)

Initialization:
- $x_i^0 \in \mathbb{R}^{n_i}$, $y_i^0 \in \mathbb{R}^m \quad \forall i \in \mathcal{V}$
- $q_i^0 \leftarrow 0 \quad \forall i \in \mathcal{V}$
- $k \leftarrow 0$

do in parallel $\forall i \in \mathcal{V}$
- $x_i^{k+1} \leftarrow$ Procedure (36)
- $y_i^{k+1} \leftarrow$ Procedure (29)
- $q_i^{k+1} \leftarrow$ Procedure (27)
- $k \leftarrow k+1$

while *not converged or stopping criterion is not met*;

IX. Simulations

We examine the performance of D-RAMS in distributed resource allocation problems, comparing it to the distributed methods DDA [12], C-SP-SG [13], and DPDA [14], in addition to examining the convergence rate of D-RAMS-Ax. We begin by examining each method on problems without local constraints before considering the performance of each method on problems where each agent has local constraints only accessible to it. We use Metropolis-Hastings weights in DDA which requires a symmetric doubly-stochastic weighting matrix and select the parameters that provide fast convergence for each method. We examine the convergence rate of each method to the optimal solution obtained from a centralized method.

A. Resource Allocation with No Local Constraints

We consider the resource allocation problem among a group of N agents with no local constraints, given by

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \sum_{i=1}^{N} \| G_i x_i - p_i \|_{\mathcal{W}_i}^2 \\ \text{subject to} \quad & \sum_{i=1}^{N} (C_i x_i - d_i) \leq 0 \end{aligned} \tag{42}$$

where $x_i \in \mathbb{R}^{n_i}$, $C_i \in \mathbb{R}^{m \times n_i}$, and $d_i \in \mathbb{R}^m$. The objective function specifies the desire of agent i to select an allocation which satisfies its tasks in $p_i \in \mathbb{R}^{a_i}$ which depend on $G_i \in \mathbb{R}^{a_i \times n_i}$, with $\mathcal{W}_i \in \mathbb{R}^{a_i \times a_i}$ indicating the relative importance of each task. The affine coupling constraint in (42) represents the limits on the availability of the resources. We examine the performance of each method in solving (42) for a small group of agents with $N = 12$ agents and a larger group of agents with $N = 50$ agents. For D-RAMS-Ax, we utilize the update procedure in (36) for computing the resource allocation of each agent, where each agent uses the exact Hessian given by (40). D-RAMS-Ax enables each agent to update its primal variables in closed-form, without resorting to a nested iterative procedure for solving its local optimization problems, resulting in greater computation efficiency.

With $N = 12$ agents, $n_i = 9$ resources $\forall i \in \mathcal{V}$, and $m = 13$, we consider a random connected communication graph with a connectivity ratio κ of 0.546, where the connectivity ratio represents the proportion of edges in the graph compared to the total number of possible edges $\kappa = \frac{2|\mathcal{E}|}{N(N-1)}$, with an average of 3 edges incident on each agent. In Figure 1, we examine the convergence of the iterates of each method to the optimal solution. Notably, the convergence performance of D-RAMS-Ax overlaps that of D-RAMS, with each algorithm converging at the same rate. D-RAMS and D-RAMS-Ax achieve the fastest convergence rate and converge in about 170 iterations, while DPDA requires more than 10^4 iterations for convergence, converging faster than DDA and C-SP-SG. In addition, the objective value of the iterates in D-RAMS and D-RAMS-Ax converge faster than all the other methods to the optimal objective value, depicted in Figure 2, with both methods achieving the same convergence rate. The objective value of the iterates in DPDA converges faster than DDA and C-SP-SG. In Figure 3, we examine the convergence rate of the value of the coupling constraint to its optimal value for each method. With D-RAMS and D-RAMS-Ax, the value of the coupling constraints converge in about 170 iterations, faster than all other methods which require more than 10^4 iterations for convergence, with the convergence rate of D-RAMS-Ax overlapping that of D-RAMS.

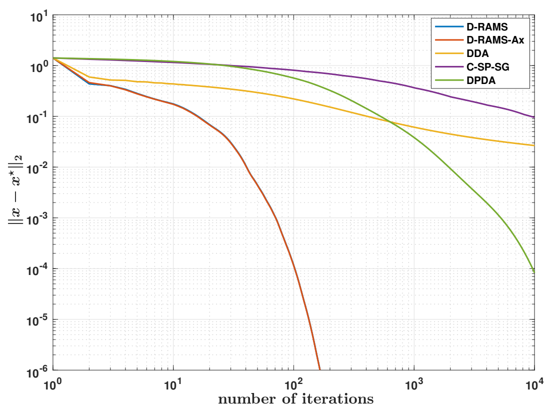

Fig. 1. Convergence rate of the solution obtained by D-RAMS, D-RAMS-Ax, DDA, C-SP-SG, and DPDA to the optimal solution of the problem in (42) with $N = 12$ agents. D-RAMS and D-RAMS-Ax attain the fastest convergence rate to the optimal solution, converging in about **170** iterations, with the convergence rate of D-RAMS-Ax overlapping that of D-RAMS. Meanwhile, DPDA converges faster than DDA and C-SP-SG.

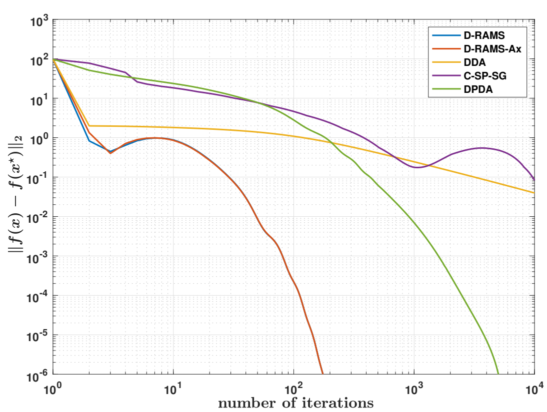

Fig. 2. Convergence rate of the objective value obtained by D-RAMS, D-RAMS-Ax, DDA, C-SP-SG, and DPDA to the optimal objective value of the problem in (42) with $N = 12$ agents. The convergence rate of D-RAMS overlaps with that of D-RAMS-Ax. D-RAMS and D-RAMS-Ax converge in about **170** iterations, attaining the fastest convergence rate. DPDA converges in about **4950** iterations, faster than DDA and C-SP-SG.

We examine the convergence rates of each method for $N = 50$ agents, $n_i = 30$ resources $\forall i \in \mathcal{V}$, and $m = 22$ with a connectivity ratio κ of 0.327 and an average of 8 edges incident

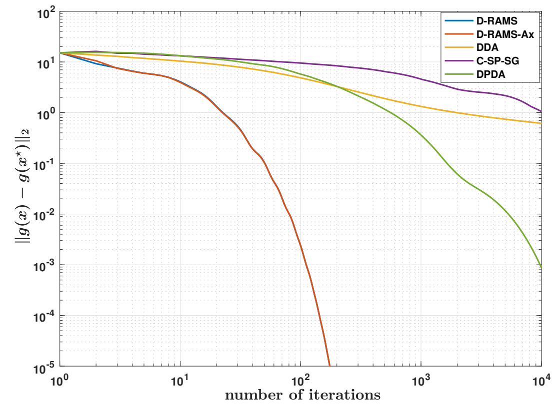

Fig. 3. Convergence rate of the value of the coupling constraint to its optimal value in (42) with $N = 12$ agents for D-RAMS, D-RAMS-Ax, DDA, C-SP-SG, and DPDA. The convergence rate of D-RAMS overlaps with that of D-RAMS-Ax, with the value of the coupling constraint in both methods converging in about **170** iterations, while the other methods require more than 10^4 iterations for convergence.

on each agent. D-RAMS and D-RAMS-Ax exhibit the same convergence performance, depicted in Figure 4. With D-RAMS and D-RAMS-Ax, the iterates converge to the optimal solution in about 220 iterations, attaining the fastest convergence rate compared to all other methods which require more than 10^4 iterations for convergence, with DDA converging faster than DPDA and C-SP-SG. Likewise, the objective value of the iterates in D-RAMS and D-RAMS-Ax converge faster than all other methods in about 250 iterations, in Figure 5. Moreover, the value of the coupling constraint in D-RAMS and D-RAMS-Ax converge to its optimal value in about 260 iterations, faster than all other methods in Figure 6, with the convergence rate of D-RAMS-Ax overlapping that of D-RAMS. Notably, while the convergence rate of DDA, C-SP-SG, and DPDA significantly declines in problems with a larger group of agents, D-RAMS and D-RAMS-Ax retain their superior convergence rates in these problems.

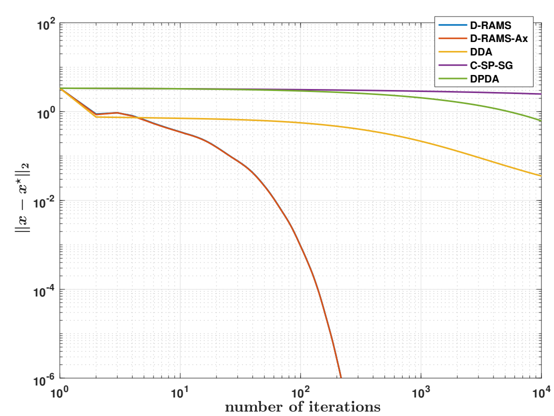

Fig. 4. Convergence rate of the iterates in D-RAMS, DDA, C-SP-SG, and DPDA to the optimal solution of (42) with $N = 50$ agents. The convergence rate of D-RAMS overlaps with that of D-RAMS-Ax, which converges in about **220** iterations, achieving the fastest convergence rate to the optimal solution. All other methods require more than 10^4 iterations for convergence, with DDA converging faster than DPDA and C-SP-SG.

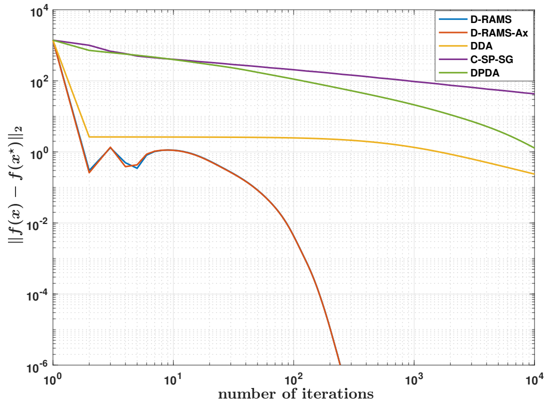

Fig. 5. Convergence rate of the objective value in D-RAMS, DDA, C-SP-SG, and DPDA to the optimal objective value of (42) with $N = 50$ agents. The convergence rate of D-RAMS overlaps with that of D-RAMS-Ax. D-RAMS and D-RAMS-Ax converge in about 250 iterations while all other methods require more than 10^4 iterations for convergence.

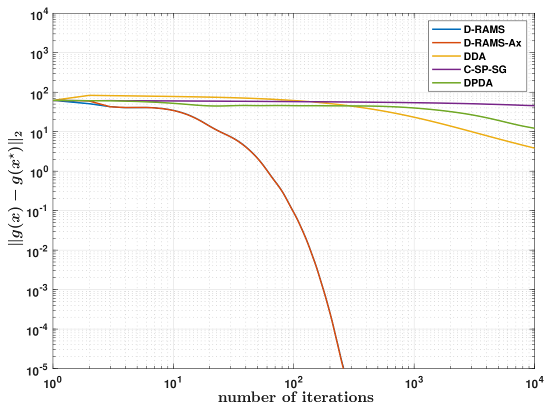

Fig. 6. Convergence rate of the value of the coupling constraint to its optimal value in (42) with $N = 50$ agents for D-RAMS, DDA, C-SP-SG, and DPDA. The convergence rate of D-RAMS overlaps with that of D-RAMS-Ax. The value of the coupling constraint in D-RAMS and D-RAMS-Ax converge in about 260 iterations while the other methods require more than 10^4 iterations for convergence.

B. Resource Allocation with Local Constraints

In these problems, each agent has a set of local constraints which is inaccessible to other agents. We consider resource allocation problems among N agents given by

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \sum_{i=1}^{N} f_i(x_i) \\ \text{subject to} \quad & \sum_{i=1}^{N} g_i(x_i) \leq 0 \\ & h_i(x_i) \leq 0 \end{aligned} \tag{43}$$

where $h_i(x_i)\colon \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{c_i}$ represents the local constraint function of agent i defined as

$$h_i(x_i) = J_i x_i - u_i \tag{44}$$

with $x_i \in \mathbb{R}^{n_i}$, $J_i \in \mathbb{R}^{a_i \times n_i}$, and $u_i \in \mathbb{R}^{a_i}$. We define the objective function of agent i as

$$f_i(x_i) = \|Q_i x_i - b_i\|_1 - \nu_i \sum_{r=1}^{n_i} \log(x_{i,r} + 1) \tag{45}$$

with $Q_i \in \mathbb{R}^{\ell_i \times n_i}$, $b_i \in \mathbb{R}^{\ell_i}$, and $\nu_i \in \mathbb{R}$. The local objective function of agent i consists of multiple measures of performance where the first component promotes the allocation of resources to satisfy its local tasks specified by Q_i and b_i. We specify the coupling constraint in (43) as

$$g_i(x_i) = \|x_i\|_{\mathcal{W}_i}^2 - d_i \tag{46}$$

with $\mathcal{W}_i \in \mathbb{R}^{n_i \times n_i}$ and $d_i \in \mathbb{R}$, signifying constraints on the availability of each resource.

We examine the performance of each method in solving (43) with $N = 30$ agents, $n_i = 50$ resources, $\ell_i = 50$, and $a_i = 46$ $\forall i \in \mathcal{V}$ on a random connected communication graph with a connectivity ratio κ of 0.559 and an average of about 8 edges incident on each agent. In D-RAMS-Ax, we utilize the primal update procedure using the exact Hessian given by (36) and (40). The iterates in D-RAMS converge in about 70 iterations to the optimal solution, achieving the fastest convergence rate compared to all other methods in Figure 7. With D-RAMS-Ax, the iterates converge in about 100 iterations, faster than DPDA, C-SP-SG, and DDA. Likewise, the objective value in D-RAMS converges faster than all other methods to the optimal objective value. D-RAMS requires 90 iterations for convergence while D-RAMS-Ax converges within 115 iterations, with all other methods requiring more than 10^3 iterations for convergence, depicted in Figure 8. In addition, while the value of the coupling constraint in DPDA, C-SP-SG, and DDA has not converged after 10^3 iterations in Figure 9, the value of the coupling constraint in D-RAMS and D-RAMS-Ax converges to its optimal value in about 110 and 140 iterations respectively. The DDA and C-SP-SG algorithms converge at a significantly slower rate compared to the other distributed algorithms, as reflected in Figure 8.

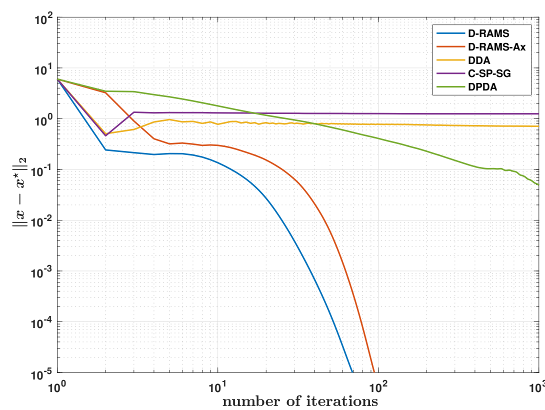

Fig. 7. Convergence rate of the iterates in D-RAMS, D-RAMS-Ax, DDA, C-SP-SG, and DPDA to the optimal solution of (43) with $N = 30$ agents. D-RAMS achieves the fastest convergence rate to the optimal solution, converging in about 70 iterations. The iterates in D-RAMS-Ax converge within 100 iterations, while all other methods require more than 10^3 iterations for convergence, with DPDA converging faster than DDA and C-SP-SG.

X. Conclusion

We derive a distributed resource allocation method where each agent communicates locally with its neighbors to compute a resource allocation which satisfies the coupling constraint on

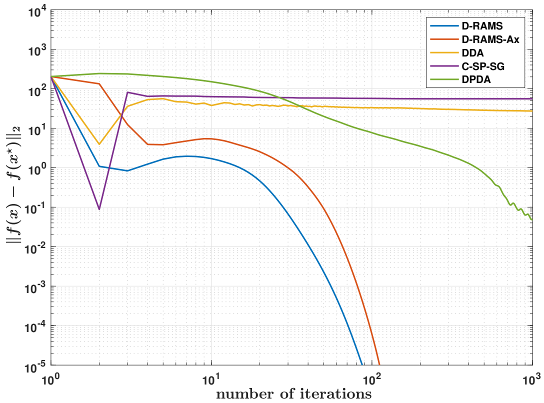

Fig. 8. Convergence rate of the objective value in D-RAMS, D-RAMS-Ax, DDA, C-SP-SG, and DPDA to the optimal objective value of (43) with $N = 30$ agents. While all other methods require more than 10^3 iterations for convergence, D-RAMS and D-RAMS-Ax converge in about 90 and 115 iterations respectively.

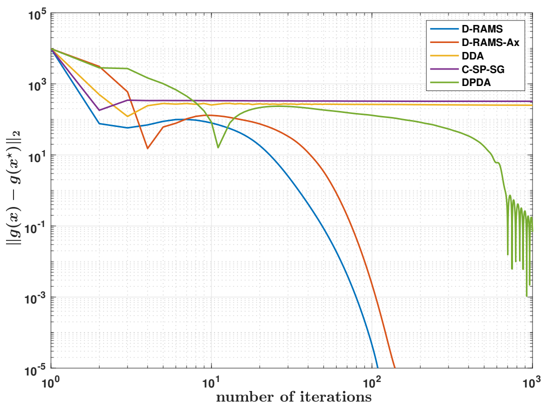

Fig. 9. Convergence rate of the value of the coupling constraint to its optimal value in (43) with $N = 30$ agents for D-RAMS, D-RAMS-Ax, DDA, C-SP-SG, and DPDA. In D-RAMS, the value of the coupling constraint converges in about 110 iterations, while D-RAMS-Ax requires about 140 iterations for convergence, with all other methods requiring greater than 10^3 iterations for convergence.

the resource allocation of all agents. Each agent neither knows nor computes the resource allocation of other agents, preserving the privacy of each agent's allocation. Our method provides linear convergence of the resource allocation and the associated dual iterates of each agent to the optimal resource allocation and dual solution respectively, attaining a faster convergence rate compared to other distributed methods. In addition, our algorithm retains its faster convergence rates in problems with larger networks of agents, over different communication graph topology, as demonstrated in our simulations.

Appendix

Derivation of the Update Procedures

We derive the update procedures for y_i along with the updates for the slack variables and Lagrange multipliers. Updating the slack variables as the maximizer of the augmented Lagrangian in (20) results in

$$\alpha_{ij}^{k+1} = \frac{u_{ij}^k + w_{ij}^k}{2\rho} + \frac{1}{2}\left(y_i^{k+1} + y_j^{k+1}\right) \tag{47}$$

at iteration k with the same update procedure for γ_{ij}. Through gradient descent on the augmented Lagrangian, each agent updates its local Lagrange multipliers using

$$u_{ij}^{k+1} = u_{ij}^k - \rho\left(\alpha_{ij}^{k+1} - y_i^{k+1}\right) \tag{48}$$

with a similar procedure for w_{ij} given by

$$w_{ij}^{k+1} = w_{ij}^k - \rho\left(\gamma_{ij}^{k+1} - y_j^{k+1}\right) \tag{49}$$

after the updates to $\boldsymbol{y}$ at iteration k. By applying (47) in (48), the update for u_{ij} simplifies to

$$u_{ij}^{k+1} = \frac{1}{2}\left(u_{ij}^k - w_{ij}^k\right) + \frac{\rho}{2}\left(y_i^{k+1} - y_j^{k+1}\right) \tag{50}$$

at iteration k. If $u_{ij}^k = -w_{ij}^k$ at any iteration k, then $u_{ij}^{k+1} = -w_{ij}^{k+1}$. Hence at each iteration k, $u_{ij}^k = -w_{ij}^k$ by initializing u_{ij}^0 and w_{ij}^0 to zero, resulting in the update procedures in (22), (23), and (24).

From the existence of a saddle-point, we can swap the order of the optimization problems in (28), resulting in

$$\begin{aligned}\underset{x_i\in\mathcal{X}_i}{\text{minimize}}\ \underset{y_i\in\mathcal{Y}}{\text{maximize}}\ \Bigg\{ & f_i(x_i) + y_i^\mathsf{T} g_i(x_i) - q_i^{k\mathsf{T}} y_i \\ & - \rho \sum_{j\in\mathcal{N}_i}\left\| y_i - \frac{y_i^k + y_j^k}{2}\right\|_2^2 \Bigg\}\end{aligned} \tag{51}$$

with the inner problem involving maximization over y_i. We can express the optimization problem in (51) as

$$\begin{aligned}&\underset{x_i\in\mathcal{X}_i}{\text{minimize}}\ \underset{y_i\in\mathcal{Y}}{\text{maximize}}\ \Bigg\{ f_i(x_i) \\ &+ \frac{1}{4\rho|\mathcal{N}_i|}\left\| g_i(x_i) - q_i^k + \rho\sum_{j\in\mathcal{N}_i}\left(y_i^k + y_j^k\right)\right\|_2^2 \\ &- \rho|\mathcal{N}_i|\left\| y_i - \frac{1}{2\rho|\mathcal{N}_i|}\left(g_i(x_i) - q_i^k + \rho\sum_{j\in\mathcal{N}_i}\left(y_i^k + y_j^k\right)\right)\right\|_2^2 \\ &- \frac{\rho}{4}\sum_{j\in\mathcal{N}_i}\left\| y_i^k + y_j^k\right\|_2^2 \Bigg\}\end{aligned} \tag{52}$$

which enables us to compute a saddle-point for (51) given by the closed-form solution for y_i^{k+1} and x_i^{k+1} in (29) and (31).

Using the closed-form solution for y_i at iteration k:

$$y_i^{k+1} = \frac{1}{2\rho|\mathcal{N}_i|}\max\left(0,\ \psi_i^k\left(x_i^{k+1}\right)\right) \tag{53}$$

with

$$\psi_i^k(x_i) = g_i(x_i) - q_i^k + \rho\sum_{j\in\mathcal{N}_i}\left(y_i^k + y_j^k\right), \tag{54}$$

the minimization problem for x_i^{k+1} simplifies to

$$\begin{aligned}\underset{x_i\in\mathcal{X}_i}{\text{minimize}}\ \Bigg\{ & f_i(x_i) + \frac{1}{4\rho|\mathcal{N}_i|}\left\|\psi_i^k(x_i)\right\|_2^2 \\ & - \rho|\mathcal{N}_i|\left\| -\frac{1}{2\rho|\mathcal{N}_i|}\min(0, \psi_i^k(x_i))\right\|_2^2 \Bigg\},\end{aligned} \tag{55}$$

which reduces to

$$\underset{x_i\in\mathcal{X}_i}{\text{minimize}}\left\{f_i(x_i)+\frac{1}{4\rho|\mathcal{N}_i|}\left\|\psi_i^k(x_i)\right\|_2^2 - \frac{1}{4\rho|\mathcal{N}_i|}\left\|\min(0,\psi_i^k(x_i))\right\|_2^2\right\}. \tag{56}$$

Further simplification of the minimization problem results in

$$\underset{x_i\in\mathcal{X}_i}{\text{minimize}}\left\{f_i(x_i)+\frac{1}{4\rho|\mathcal{N}_i|}\Big(\left\|\psi_i^k(x_i)\right\|_2^2 - \left\|\min(0,\psi_i^k(x_i))\right\|_2^2\Big)\right\}. \tag{57}$$

As such, agent i computes its resource allocation from the problem

$$\underset{x_i\in\mathcal{X}_i}{\text{minimize}}\left\{f_i(x_i)+\frac{1}{4\rho|\mathcal{N}_i|}\left\|\max(0,\psi_i^k(x_i))\right\|_2^2\right\} \tag{58}$$

at iteration k.

Proof of Theorem 2

We introduce the sequence $\{z^k\}$, consisting of the slack variable α^k and the Lagrange multiplier u^k, as well as the W-weighted norm $\|\cdot\|_W^2$, with

$$z^k=\begin{bmatrix}u^k\\ \alpha^k\end{bmatrix},\quad W=\begin{bmatrix}\frac{1}{\rho} & 0\\ 0 & \rho\end{bmatrix}, \tag{59}$$

where $W>0$. We make use of the following lemma in proving linear convergence of $\{\boldsymbol{y}^k\}$. The proof follows the same procedure in [42]. Consequently, we omit the proof here.

Lemma 3. *The error in the slack variables and Lagrange multipliers $\left\|z^k-z^\star\right\|_W^2$ decreases monotonically at each iteration, ultimately converging to zero, with*

$$\left\|z^k-z^\star\right\|_W^2-\left\|z^{k+1}-z^\star\right\|_W^2 \geq \left\|z^{k+1}-z^k\right\|_W^2+\lambda\left\|\boldsymbol{y}^{k+1}-\boldsymbol{y}^\star\right\|_2^2, \tag{60}$$

where $z^\star$ denotes an optimal solution for z^k and $\lambda>0$.

Moreover, the sequence $\{z^k\}$ converges Q-linearly to $z^\star$, such that

$$\frac{\left\|z^{k+1}-z^\star\right\|_W^2}{\left\|z^k-z^\star\right\|_W^2}\leq\frac{1}{\zeta+1} \tag{61}$$

where

$$\zeta=\min\left\{\frac{4\lambda\rho(\mu-1)\sigma_{\min}^2(\mathcal{E}_l-\mathcal{E}_r)}{\rho^2\sigma_{\max}^2(\mathcal{E}_l+\mathcal{E}_r)\sigma_{\min}^2(\mathcal{E}_l-\mathcal{E}_r)(\mu-1)+4\mu L},\ \frac{\sigma_{\min}^2(\mathcal{E}_l-\mathcal{E}_r)}{\mu\sigma_{\min}^2(\mathcal{E}_l+\mathcal{E}_r)}\right\} \tag{62}$$

with $\zeta>0$, $L>0$, and $\mu>1$.

The matrices $\mathcal{E}_l$ and $\mathcal{E}_r$ are derived from the communication graph $\mathcal{G}$ with blocks $\mathcal{E}_{l,(k,i)}$ and $\mathcal{E}_{r,(k,j)}$ set to the identity matrix $I\in\mathbb{R}^{m\times m}$ for an edge k from agent i to agent j. From Q-linear convergence of $\{z^k\}$ and (60), in Lemma 3,

$$\left\|\boldsymbol{y}^{k+1}-\boldsymbol{y}^\star\right\|_2^2\leq\frac{1}{\lambda}\left\|z^k-z^\star\right\|_W^2 \tag{63}$$

which shows R-linear convergence of the dual iterates $\{\boldsymbol{y}^k\}$ to the optimal dual solution $\boldsymbol{y}^\star$.

Dependence of the Convergence Rate on the Communication Network

The definition of ζ in (62) depends on the value of $\sigma_{\min}^2(\mathcal{E}_l-\mathcal{E}_r)$, the minimum non-zero singular value of the matrix $(\mathcal{E}_l-\mathcal{E}_r)$, which is related to the graph Laplacian L through the relationship

$$L=\frac{1}{2}(\mathcal{E}_l-\mathcal{E}_r)^\top(\mathcal{E}_l-\mathcal{E}_r)\ , \tag{64}$$

indicating symmetry and positive semi-definiteness of the graph Laplacian. The graph Laplacian has been studied extensively due to its relevance to the connectedness of graphs and the enumeration of spanning trees [43], [44]. Notably, the smallest non-zero singular value of the graph Laplacian, also referred to as the algebraic connectivity or Fielder value of the graph, is a measure of the connectedness of the graph. Since the minimum non-zero singular value of the matrix $(\mathcal{E}_l-\mathcal{E}_r)$ corresponds to the square root of the minimum non-zero singular value of L, the value of ζ in (62) depends on the algebraic connectivity of the underlying communication graph. Consequently, the topology of the communication graph $\mathcal{G}$ influences the convergence rate of the dual iterates to the optimal dual solution. In general, a graph with a larger algebraic connectivity would result in a faster convergence rate.

Proof of Theorem 3

Since a saddle-point exists for the Lagrangian of (5), an optimal resource allocation $\boldsymbol{x}^\star$ and a corresponding optimal dual solution $y^\star$ satisfy the Karush-Kuhn-Tucker (KKT) optimality conditions. As a result,

$$\nabla f(\boldsymbol{x}^\star)+g(\boldsymbol{x}^\star)^\top y^\star=0 \tag{65}$$

where, in addition, $\boldsymbol{x}^\star$ lies in the feasible set $\boldsymbol{\mathcal{X}}$. The update procedure for x_i^k in (31) indicates that

$$\nabla f_i(x_i^{k+1})+g_i(x_i^{k+1})^\top y_i^{k+1}=0 \tag{66}$$

at each iteration k, where x_i^{k+1} satisfies the local constraints of agent i. From Theorem 2, the local dual variable of each agent converges to the optimal dual solution; hence, the resource allocation of agent i satisfies

$$\nabla f_i(x_i^s)+g_i(x_i^s)^\top y^\star=0, \tag{67}$$

where x_i^s represents the resource allocation of agent i computed upon convergence of its dual variables. Since the pair $(x_i^s,y^\star)$ satisfies (67), the pair $(\boldsymbol{x}^s,y^\star)$ satisfies the KKT condition in (65), as the condition in (67) is sufficient for satisfying (65). Consequently, the resource allocation $\boldsymbol{x}^s$ represents an optimal resource allocation for the problem in (5). Hence, the sequence of resource allocation of each agent converges to an optimal resource allocation.

Assuming strong convexity of the objective and coupling constraint functions, we prove stronger results on the convergence rate of the sequence of resource allocations. The optimality conditions of $\boldsymbol{x}$ indicate that

$$\left\|\nabla f(\boldsymbol{x}^{k+1}) - \nabla f(\boldsymbol{x}^\star)\right\|_2^2 \leq \sigma_{\max}^2(J_g)\left\|\boldsymbol{y}^{k+1} - \boldsymbol{y}^\star\right\|_2^2 \quad (68)$$

at each iteration k, where J_g denotes the Jacobian of the coupling constraint between the agents. Provided the objective function $f(\boldsymbol{x})$ is $\bar{\nu}$-strongly convex with $\bar{\nu} > 1$,

$$\left(\nabla f(\boldsymbol{x}^{k+1}) - \nabla f(\boldsymbol{x}^\star)\right)^\mathsf{T}\left(\boldsymbol{x}^{k+1} - \boldsymbol{x}^\star\right) \geq \bar{\nu}\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^\star\right\|_2^2 \quad (69)$$

for any pair of resource allocations $\boldsymbol{x}^{k+1}$ and $\boldsymbol{x}^\star$. Using

$$\begin{aligned}\left\|\nabla f(\boldsymbol{x}^{k+1}) - \nabla f(\boldsymbol{x}^\star)\right\|_2^2 + \left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^\star\right\|_2^2 \\ \geq \left(\nabla f(\boldsymbol{x}^{k+1}) - \nabla f(\boldsymbol{x}^\star)\right)^\mathsf{T}\left(\boldsymbol{x}^{k+1} - \boldsymbol{x}^\star\right)\end{aligned} \quad (70)$$

with (69) results in

$$\left\|\nabla f(\boldsymbol{x}^{k+1}) - \nabla f(\boldsymbol{x}^\star)\right\|_2^2 \geq \nu\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^\star\right\|_2^2 \quad (71)$$

with $\nu > 0$.

From (68) and (71),

$$\sigma_{\max}^2(J_g)\left\|\boldsymbol{y}^{k+1} - \boldsymbol{y}^\star\right\|_2^2 \geq \nu\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^\star\right\|_2^2 \quad (72)$$

and

$$\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^\star\right\|_2^2 \leq \frac{\sigma_{\max}^2(J_g)}{\nu}\left\|\boldsymbol{y}^{k+1} - \boldsymbol{y}^\star\right\|_2^2, \quad (73)$$

showing R-linear convergence of $\{\boldsymbol{x}^k\}$ to $\boldsymbol{x}^\star$ which follows from R-linear convergence of $\{\boldsymbol{y}^k\}$ to $\boldsymbol{y}^\star$.

Proof of Theorem 4

The violation of the coupling constraint at iteration k

$$\left\|g(\boldsymbol{x}^{k+1})\right\|_2^2 = \left\|g(\boldsymbol{x}^{k+1}) - g(\boldsymbol{x}^\star)\right\|_2^2, \quad (74)$$

from feasibility of the optimal resource allocation $\boldsymbol{x}^\star$, for problems with an equality coupling constraint. With an affine constraint function,

$$\left\|g(\boldsymbol{x}^{k+1}) - g(\boldsymbol{x}^\star)\right\|_2^2 \leq \sigma_{\max}^2(\mathcal{I}^\mathsf{T}J_g)\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^\star\right\|_2^2 \quad (75)$$

where $\mathcal{I} \in \mathbb{R}^{Nm \times m}$ represents the vertical concatenation of N copies of the identity matrix. Combining (74) and (75) results in

$$\left\|g(\boldsymbol{x}^{k+1})\right\|_2^2 \leq \sigma_{\max}^2(\mathcal{I}^\mathsf{T}J_g)\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^\star\right\|_2^2 \quad (76)$$

at iteration k. Hence, the violation of the coupling constraint on the resource allocations converges R-linearly to zero, following from R-linear convergence of $\{\boldsymbol{x}^k\}$ to $\boldsymbol{x}^\star$.

References

[1] M. Belleschi, G. Fodor, and A. Abrardo, "Performance analysis of a distributed resource allocation scheme for d2d communications," in *2011 ieee globecom workshops (gc wkshps)*. IEEE, 2011, pp. 358–362.

[2] D. A. Schmidt, C. Shi, R. A. Berry, M. L. Honig, and W. Utschick, "Distributed resource allocation schemes," *IEEE Signal Processing Magazine*, vol. 26, no. 5, pp. 53–63, 2009.

[3] D. Gesbert, S. G. Kiani, A. Gjendemsjo, and G. E. Oien, "Adaptation, coordination, and distributed resource allocation in interference-limited wireless networks," *Proceedings of the IEEE*, vol. 95, no. 12, pp. 2393–2409, 2007.

[4] H. Halabian, "Distributed resource allocation optimization in 5g virtualized networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 627–642, 2019.

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

[6] Y. Zhu, W. Ren, W. Yu, and G. Wen, "Distributed resource allocation over directed graphs via continuous-time algorithms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 2, pp. 1097–1106, 2019.

[7] G. Chen, J. Ren, and E. N. Feng, "Distributed finite-time economic dispatch of a network of energy resources," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 822–832, 2016.

[8] A. Nedić, A. Olshevsky, and W. Shi, "Improved convergence rates for distributed resource allocation," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 172–177.

[9] Z. Deng, S. Liang, and Y. Hong, "Distributed continuous-time algorithms for resource allocation problems over weight-balanced digraphs," *IEEE transactions on cybernetics*, vol. 48, no. 11, pp. 3116–3125, 2017.

[10] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2011.

[11] T.-H. Chang, A. Nedić, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1524–1538, 2014.

[12] A. Falsone, K. Margellos, S. Garatti, and M. Prandini, "Dual decomposition for multi-agent distributed optimization with coupling constraints," *Automatica*, vol. 84, pp. 149–158, 2017.

[13] D. Mateos-Núnez and J. Cortés, "Distributed saddle-point subgradient algorithms with laplacian averaging," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2720–2735, 2016.

[14] N. S. Aybat and E. Y. Hamedani, "A distributed admm-like method for resource sharing over time-varying networks," *SIAM Journal on Optimization*, vol. 29, no. 4, pp. 3036–3068, 2019.

[15] B. Johansson and M. Johansson, "Distributed non-smooth resource allocation over a network," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, 2009, pp. 1678–1683.

[16] H. Lakshmanan and D. P. De Farias, "Decentralized resource allocation in dynamic networks of agents," *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 911–940, 2008.

[17] T. T. Doan and A. Olshevsky, "Distributed resource allocation on dynamic networks in quadratic time," *Systems & Control Letters*, vol. 99, pp. 57–63, 2017.

[18] Y. Xu, T. Han, K. Cai, Z. Lin, G. Yan, and M. Fu, "A distributed algorithm for resource allocation over dynamic digraphs," *IEEE Transactions on Signal Processing*, vol. 65, no. 10, pp. 2600–2612, 2017.

[19] T. T. Doan and C. L. Beck, "Distributed lagrangian methods for network resource allocation," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2017, pp. 650–655.

[20] A. Beck, A. Nedić, A. Ozdaglar, and M. Teboulle, "An $o(1/k)$ gradient method for network resource allocation problems," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 64–73, 2014.

[21] J. Zhang, K. You, and K. Cai, "Distributed dual gradient tracking for resource allocation in unbalanced networks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2186–2198, 2020.

[22] 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.

[23] B. Johansson and M. Johansson, "Primal and dual approaches to distributed cross-layer optimization," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 113–118, 2005.

[24] L. Su, M. Li, V. Gupta, and G. Chesi, "Distributed resource allocation over time-varying balanced digraphs with discrete-time communication," *IEEE Transactions on Control of Network Systems*, 2021.

[25] S. S. Kia, "An augmented lagrangian distributed algorithm for an in-network optimal resource allocation problem," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 3312–3317.

[26] D. Niu and B. Li, "An efficient distributed algorithm for resource allocation in large-scale coupled systems," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 1501–1509.

[27] B. Zhang, C. Gu, and J. Li, "Distributed convex optimization with coupling constraints over time-varying directed graphs," *Journal of Industrial & Management Optimization*, vol. 17, no. 4, p. 2119, 2021.

[28] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5262–5276, 2010.

[29] T.-H. Chang, "A proximal dual consensus admm method for multi-agent constrained optimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3719–3734, 2016.

[30] G. Banjac, F. Rey, P. Goulart, and J. Lygeros, "Decentralized resource allocation via dual consensus admm," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 2789–2794.

[31] A. Falsone, I. Notarnicola, G. Notarstefano, and M. Prandini, "Tracking-admm for distributed constraint-coupled optimization," *Automatica*, vol. 117, p. 108962, 2020.

[32] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "A dual splitting approach for distributed resource allocation with regularization," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 1, pp. 403–414, 2018.

[33] S. Liang, X. Zeng, and Y. Hong, "Distributed sub-optimal resource allocation over weight-balanced graph via singular perturbation," *Automatica*, vol. 95, pp. 222–228, 2018.

[34] Y. Chen, G. Lan, and Y. Ouyang, "Optimal primal-dual methods for a class of saddle point problems," *SIAM Journal on Optimization*, vol. 24, no. 4, pp. 1779–1814, 2014.

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

[36] A. Chambolle and T. Pock, "On the ergodic convergence rates of a first-order primal–dual algorithm," *Mathematical Programming*, vol. 159, no. 1, pp. 253–287, 2016.

[37] D. GABAY, "Applications of the method of multipliers to variational inequalities," *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, pp. 299–331, 1983.

[38] L. Chen, D. Sun, and K.-C. Toh, "A note on the convergence of admm for linearly constrained convex optimization problems," *Computational Optimization and Applications*, vol. 66, no. 2, pp. 327–343, 2017.

[39] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970, vol. 36.

[40] J.-B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms I: Fundamentals*. Springer Science & Business Media, 1996, vol. 305.

[41] J. E. Dennis, Jr and J. J. Moré, "Quasi-newton methods, motivation and theory," *SIAM review*, vol. 19, no. 1, pp. 46–89, 1977.

[42] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," RICE UNIV HOUSTON TX DEPT OF COMPUTATIONAL AND APPLIED MATHEMATICS, Tech. Rep., 2012.

[43] M. Fiedler, "Laplacian of graphs and algebraic connectivity," *Banach Center Publications*, vol. 25, no. 1, pp. 57–70, 1989.

[44] N. M. M. De Abreu, "Old and new results on algebraic connectivity of graphs," *Linear algebra and its applications*, vol. 423, no. 1, pp. 53–73, 2007.