Annual Reviews in Control 59 (2025) 100983

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

Review article

# Survey of distributed algorithms for resource allocation over multi-agent systems

Mohammadreza Doostmohammadian [a,*], Alireza Aghasi [b], Mohammad Pirani [c], Ehsan Nekouei [d], Houman Zarrabi [e], Reza Keypour [f], Apostolos I. Rikos [g], Karl H. Johansson [h]

[a] Faculty of Mechanical Engineering, Semnan University, Semnan, Iran
[b] Department of Electrical Engineering and Computer Science, Oregon State University, USA
[c] Department of Mechanical Engineering, University of Ottawa, Ottawa, ON, Canada
[d] City University of Hong Kong, Hong Kong
[e] Iran Telecom Research Center (ITRC), Tehran, Iran
[f] Faculty of Electrical Engineering, Semnan University, Semnan, Iran
[g] Artificial Intelligence Thrust of the Information Hub, Hong Kong University of Science and Technology (Guangzhou), China
[h] Division of Decision and Control Systems, KTH Royal Institute of Technology, Sweden

ARTICLE INFO

*Keywords:*
Distributed coupling-constrained optimization
Distributed resource allocation
Graph theory
Consensus
Convex analysis

ABSTRACT

Resource allocation and scheduling in multi-agent systems present challenges due to complex interactions and decentralization. This survey paper provides a comprehensive analysis of distributed algorithms for addressing the distributed resource allocation (DRA) problem over multi-agent systems. It covers a significant area of research at the intersection of optimization, multi-agent systems, and distributed consensus-based computing. The paper begins by presenting a mathematical formulation of the DRA problem, establishing a solid foundation for further exploration. Real-world applications of DRA in various domains are examined to underscore the importance of efficient resource allocation, and relevant distributed optimization formulations are presented. The survey then delves into existing solutions for DRA, encompassing linear, nonlinear, primal-based, and dual-formulation-based approaches. Furthermore, this paper evaluates the features and properties of DRA algorithms, addressing key aspects such as feasibility, convergence rate, and network reliability. The analysis of mathematical foundations, diverse applications, existing solutions, and algorithmic properties contributes to a broader comprehension of the challenges and potential solutions for this domain.

**Contents**



* Corresponding author.
*E-mail addresses:* doost@semnan.ac.ir (M. Doostmohammadian), alireza.aghasi@oregonstate.edu (A. Aghasi), mpirani@uottawa.ca (M. Pirani), enekouei@cityu.edu.hk (E. Nekouei), h.zarrabi@itrc.ac.ir (H. Zarrabi), rkeypour@semnan.ac.ir (R. Keypour), apostolosr@hkust-gz.edu.cn (A.I. Rikos), kallej@kth.se (K.H. Johansson).

https://doi.org/10.1016/j.arcontrol.2024.100983
Received 22 January 2024; Received in revised form 7 August 2024; Accepted 18 November 2024
Available online 29 November 2024

## 1. Introduction

In today's technological landscape, decentralized architectures are on the rise. Distributed algorithms are gaining increasing importance, thanks to their compatibility with various distributed platforms like the Internet of Things (IoT), cloud-based computing, and parallel processors. Many real-world systems, such as computer networks, peer-to-peer networks, cloud computing environments, and blockchain networks, inherently operate distributedly. Distributed algorithms serve as the cornerstone for designing efficient and scalable solutions in these systems. By studying and developing these algorithms, we can address practical challenges, enhance system performance, and improve functionality. Furthermore, distributed algorithms offer fault-tolerant solutions by distributing workload and data across multiple nodes/agents,[1] mitigating the single-point-of-failure issue. In the event of a node failure or malfunction, the system can properly redistribute tasks to the remaining nodes, ensuring continued operation. This fault-tolerance property ensures that the system remains operational and resilient in the face of failures, leading to increased reliability and availability (Dibaji et al., 2019). Large-scale applications range from distributed estimation and monitoring of dynamical systems (Doostmohammadian & Khan, 2013; Kar & Moura, 2013; Park & Martins, 2012), distributed coordination among robots for cooperative tasks (Jadbabaie, Lin, & Morse, 2003), fault-detection in dynamical systems (Pasqualetti, Dorfler, & Bullo, 2013) to more recent optimization, machine learning and distributed data-mining solutions (Khan, Bajwa, Nedić, Rabbat, & Sayed, 2020) and power flow optimization in smart grids (Cherukuri & Cortés, 2015; Hadjicostis & Charalambous, 2011; Molzahn et al., 2017; Yang, Tan, & Xu, 2013).

As a motivating example consider a micro-grid with a sustainable energy system by integrating distributed energy resources, see Fig. 1.

This grid consists of various energy resources, including solar panels, wind turbines, battery storage systems, and PEV (plug-in electric vehicle) charging stations. The goal is to optimize the use of these resources to ensure reliable energy supply, minimize costs, and reduce carbon emissions. The main challenge is how to locally manage the distributed energy resources available in the grid to meet the flexible demand while optimizing costs in a large-scale scenario. First, real-time data is collected from all energy generation resources, energy consumption patterns, weather forecasts, and market prices to model the cost objective and resource-demand balance (Köhler, Müller, Li, & Allgöwer, 2017). Then, a *distributed* economic dispatch algorithm is employed to determine the optimal generation and consumption schedule for each resource with no need for a centralized coordinator (Wen, Yu, & Liu, 2021). This is by considering *local* generation capacity, the capability of battery systems to store excess energy, and the demand response. An optimal solution is determined from collaborative decision-making among all the elements in the energy ecosystem in a localized approach. Benefitting from locally-defined learning and optimization techniques, the grid continuously improves its economic dispatch strategies as it learns from consumption patterns and green energy generation. Other than distributed energy resource management, there are many other applications of distributed resource allocation (DRA) including distributed CPU scheduling and task allocation, distributed automatic generation control, distributed PEV optimal charging schedule, and distributed network utility maximization. In all these applications the goal is to optimally assign the resources to reduce the cost (or to maximize the utility) in a distributed scheme and with localized decision-making. Detailed discussions on these motivating applications and more similar problems are provided in Section 6.

DRA algorithms are designed to efficiently distribute limited resources among multiple entities or agents in a distributed manner. These algorithms aim to optimize the allocation process by considering factors such as fairness (cost-optimality), efficiency, and scalability. The motivation behind these algorithms can be understood through several key aspects:

- *Scalability:* In many real-world scenarios, there are large-scale systems with numerous entities competing for limited resources. Centralized approaches may struggle to handle the computational and communication overhead to the central processor required to manage such systems. DRA algorithms address this challenge by allowing entities to make local decisions based on limited information, thus enabling scalable resource allocation in large networks (Pelikan, Sastry, & Cantú-Paz, 2006).
- *Efficiency:* Efficient utilization of resources is crucial to maximizing the overall system performance. By distributing the decision-making process across multiple entities, distributed algorithms can exploit local knowledge and adapt to dynamic conditions,

[1] In this paper, we use (computing) node and agent interchangeably.

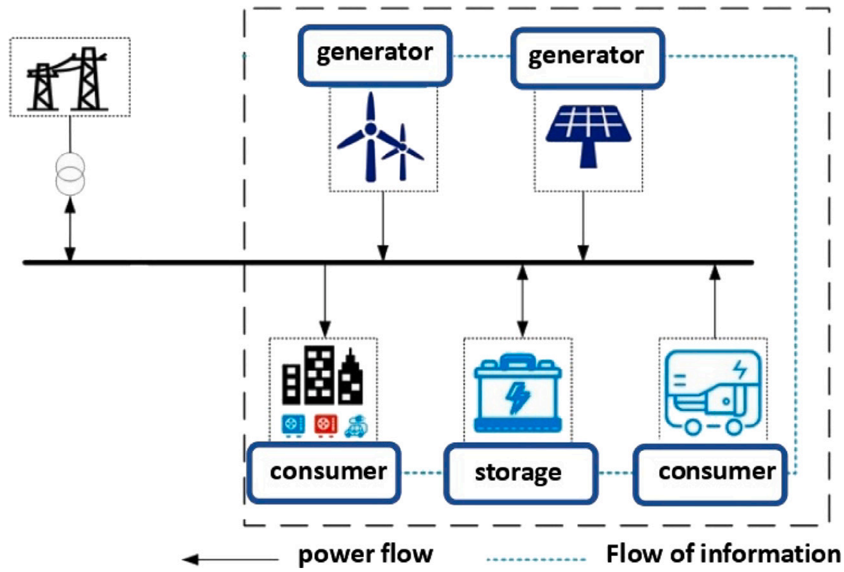


**Fig. 1.** Distributed energy resource management and economic dispatch in a smart grid with various types of electrical energy generators, storage systems, and consumers: The goal is to optimize the energy cost and manage the power flow among different entities of the smart grid network by locally exchanging information among these entities.

leading to more efficient resource allocation (Vinyals, Rodriguez-Aguilar, & Cerquides, 2011). By using multiple nodes, distributed algorithms can make better use of available resources, as each node can leverage its *local* resources more efficiently. Furthermore, distributed algorithms can facilitate decentralization and parallelizability, which is more efficient in scenarios where a central point of control may not be practical, cost-effective, or desirable, and tasks can be divided into smaller independent subtasks and processed in parallel. Examples include peer-to-peer networks and block-chain systems (Ramseyer, Goel, & Mazières, 2023).

- *Flexibility and Adaptability:* DRA algorithms are designed to be flexible and adaptable to changing environments without the intervention of a centralized entity. They can locally respond to fluctuations in resource availability or changes in agent requirements by dynamically reallocating resources (Skobelev, 2015). This adaptability is particularly valuable in dynamic and unpredictable systems, where the allocation requirements may vary over time.
- *Fault-tolerance and Robustness:* Distributed systems are often prone to failures or disruptions due to the distributed nature of the environment (Kuwaranancharoen, Xin, & Sundaram, 2020; Pirani et al., 2018). DRA algorithms can be designed to be fault-tolerant and robust, allowing them to continue functioning even in the presence of failures or network partitions. These algorithms may incorporate redundancy, consensus mechanisms, or recovery strategies to ensure reliable resource allocation.
- *Privacy and Security:* In certain scenarios, entities may have privacy concerns or security requirements that limit the sharing of sensitive information (Mo & Murray, 2016; Nekouei, Tanaka, Skoglund, & Johansson, 2019; Pequito, Kar, Sundaram, & Aguiar, 2014; Rikos, Charalambous, Johansson, & Hadjicostis, 2020; Ruan, Gao, & Wang, 2019). DRA algorithms can accommodate such constraints by allowing agents to make local decisions based on their private information or a limited set of shared information. The distributed approach reduces the exposure of sensitive data and enhances the security and privacy of the resource allocation process.

The above-mentioned motivational aspects behind DRA algorithms lie in their ability to address the challenges posed by large-scale, dynamic, and decentralized systems. By adopting local decision-making, adaptability, and feasibility considerations these algorithms achieve efficient and effective resource allocation in a distributed manner. These considerations further motivate many use-cases of DRA algorithms including wireless communication networks (Akyildiz, Su, Sankarasubramaniam, & Cayirci, 2002; Heinzelman, Chandrakasan, & Balakrishnan, 2000), distributed and cloud computing (Vlahakis, Athanasopoulos, & McLoone, 2021), traffic management (Baldacci, Mingozzi, & Roberti, 2012; Gao & Niu, 2021; Taillard, Laporte, & Gendreau, 1996), power distribution and energy management (Boqiang & Chuanwen, 2009), task assignment in multi-agent systems (Sayyaadi & Moarref, 2011), radio resource management in cellular networks (Zander, 1997), distributed sensor networks (Lesser, Ortiz, & Tambe, 2003), supply chain management (Malairajan, Ganesh, Muhos, & Anbuudayasankar, 2013), etc.

This work provides a comprehensive overview of the field of DRA and scheduling over multi-agent systems and contributes to the understanding of challenges, existing solutions, and properties of such algorithms. First and foremost, to lay a solid foundation for the subsequent analysis, a mathematical formulation of the DRA problem is presented. This formulation captures the essence of resource allocation challenges in multi-agent systems and enables a rigorous examination of potential solutions. By defining the problem in mathematical terms, along with extensive preliminaries on graph theory, consensus algorithms, and convex set analysis, our paper sets the stage for a systematic exploration of applicable distributed algorithms. Furthermore, it provides a comprehensive and up-to-date survey of the various distributed optimization techniques utilized to address the challenges posed by coupling equality constraints in multi-agent systems. Furthermore, this survey explores existing solutions and applications for the DRA problem, focusing on smart grid (automatic generation control and economic dispatch), CPU scheduling over networked data centres, plug-in electric vehicle (PEV) charging schedules, and network utility maximization (NUM). Additionally, both LG-based solutions to the primal formulation and dual-formulation-based solutions are presented. A comprehensive comparative analysis is conducted, highlighting the strengths, limitations, and applicability of each algorithm to different scenarios. This examination not only presents the state-of-the-art in DRA but also identifies the key algorithmic techniques and principles employed in these solutions. Furthermore, the paper evaluates the main features and properties of the DRA algorithms: feasibility, convergence, and network reliability. Feasibility analysis assesses the algorithms' ability to satisfy the resource-demand coupling-constraint in all iterations. Convergence analysis examines how quickly these algorithms converge to optimal or near-optimal solutions. Network reliability considerations address the robustness of algorithms in the face of communication failures, dynamic changes in the network, latency, and packet drops.

As compared to the existing literature focusing solely on distributed optimization (Molzahn et al., 2017; Nedić & Liu, 2018; Yang, Xu, Li, Liu and Huang, 2019), this work is more devoted to specifically highlighting applications and formulations of the DRA algorithms. While both types of algorithms operate in a distributed manner, general distributed optimization algorithms focus on solving global optimization problems, and DRA algorithms concentrate on efficiently allocating shared resources among multiple nodes. The choice between the two depends on the specific requirements and objectives of the distributed system in question. DRA algorithms are primarily concerned with dividing and allocating resources efficiently among multiple nodes in a distributed system with the help of *constrained* optimization algorithms. The resources could include bandwidth, computing power, storage, or any other shared resource. The goal is to ensure optimal distribution or utilization and preserve constraint-feasibility among the nodes. Therefore, applications of the DRA algorithms are often different from those of distributed optimization. In contrast to mainly machine learning applications of distributed optimization in the literature, the DRA mainly focuses on computer networks, cloud computing environments, wireless communication systems, smart grid and energy networks where efficient resource allocation is essential for maintaining system stability and performance.

By presenting a comprehensive analysis of distributed algorithms for resource allocation and scheduling over multi-agent systems, this survey paper contributes to a deeper understanding of the field. The exploration of mathematical foundations, real-world applications, existing solutions, and algorithmic properties serves as a valuable resource for researchers, practitioners, and system designers seeking to optimize resource allocation in decentralized settings. Our key contributions can be summarized as follows:

- We clearly describe the problem of resource allocation and scheduling in multi-agent systems and present its formal mathematical framework.
- We discuss various real-world applications of resource allocation and scheduling in large-scale distributed setups and networked systems.
- We survey state-of-the-art solutions for distributed resource allocation, including linear, nonlinear, primal-based, dual-formulation-based, and machine-learning-based approaches.
- We analyse different features and properties of the existing algorithms in terms of constraint-feasibility, convergence, convexity requirements, and network reliability.

*Paper Outline:* Section 2 states the DRA problem in mathematical form and Section 3 provides the preliminaries and background. Section 4 discusses different existing distributed solutions to the DRA problem and in Section 5 the main features and properties of these solutions are discussed. Section 6 presents real-world applications. Finally, Section 7 concludes the paper with some future directions.

*General Notation:* The column vectors, scalars, and matrices are respectively represented by bold small letters, small letters, and capital letters. $\partial f_i$ and $\partial^2 f_i$ denote the first derivative $\frac{d f_i(x_i)}{d x_i}$ and second derivative $\frac{d^2 f_i(x_i)}{d x_i^2}$, respectively. $\nabla F(\cdot)$ denotes the gradient of $F(\cdot)$. Vectors $\mathbf{1}_n$, $\mathbf{0}_n$ denote column vector of ones and zeros of size $n$.

## 2. The DRA problem

### 2.1. Motivational example

Consider a network of renewable energy sources, such as wind turbines and solar panels, scattered across a vast landscape. These sources generate clean, green energy, but their output is inherently intermittent and dependent on weather conditions. To maximize the utilization of these resources and ensure a reliable energy supply, a robust and adaptive resource allocation system is crucial. Traditional power grids that rely on centralized decision-making struggle to efficiently integrate renewable energy sources. Grid operators had to predict energy generation, allocate resources based on these predictions, and often relied on fossil fuel backup when renewable sources fell short, resulting in wasted energy and increased greenhouse gas emissions. Distributed resource allocation, on the other hand, introduces a game-changing approach. Each renewable energy source, equipped with sensors and intelligent control systems, can autonomously adjust its output in response to real-time weather data and energy demand. When one area experiences a sudden increase in energy demand due to, for example, an unexpected heatwave, nearby solar panels can increase their energy production, while wind turbines in windy regions can scale back during low wind forces. This distributed decision-making ensures that energy is generated where it is needed most and minimizes energy loss during transmission. Furthermore, the surplus energy generated during optimal conditions can be properly redistributed to areas with higher demand, enhancing grid resilience and reducing the need for backup power sources. Other advantages include scalability and not relying on single point of failure. Scalability implies that the algorithm is able to keep its efficiency when the system size and workload grow. No single point of failure implies that in case of losing one (or more) node/agent the rest of the networked system keeps working and performs its intended tasks (by relying on node redundancy).

Potential applications of the DRA extend beyond energy grids to various domains, including the allocation of computing resources in large-scale data centres, transportation systems, and communication networks. Such motivating applications are discussed in detail in Section 6.

### 2.2. Mathematical formulation

In DRA, the idea is to update the state of some agents $z_i$ over a network to minimize their cost function (or certain loss function), while the weighted sum of states is constant and equal to certain demand $b$. DRA in its most general form is modelled as an equality-constraint coupled optimization problem as follows,

$$\begin{aligned} \min_{\mathbf{z}} \quad & \widetilde{F}(\mathbf{z}) := \sum_{i=1}^{n} \widetilde{f}_i(z_i) \\ \text{s.t.} \quad & \mathbf{z}^\top \mathbf{a} = b, \quad \underline{z}_i \leq z_i \leq \overline{z}_i \end{aligned} \tag{1}$$

with $z_i \in \mathbb{R}$ as the state variable at agent $i$, column vector $\mathbf{z} = [z_1; \dots ; z_n] \in \mathbb{R}^n$ as the collective vector state,[2] and constraint parameters $\mathbf{a} = [a_1; \dots ; a_n] \in \mathbb{R}^n$ and $b \in \mathbb{R}$. The function $\widetilde{f}_i : \mathbb{R} \mapsto \mathbb{R}$ denotes the local objective at agent $i$, and the overall cost is $\widetilde{F} : \mathbb{R}^n \mapsto \mathbb{R}$. The box constraints $\underline{z}_i \leq z_i \leq \overline{z}_i$ are the local (non-coupling) constraints and $\mathbf{z}^\top \mathbf{a} = b$ is the coupling constraint.

Simplifying the problem described in (1), we employ a change of variables by introducing $x_i := z_i a_i$. This transformation allows us to express the problem in a different standard form commonly found in the existing literature. The simplified formulation of (1) is as follows:

$$\begin{aligned} \min_{\mathbf{x}} \quad & F(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i) \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i = b, \quad \underline{x}_i \leq x_i \leq \overline{x}_i \end{aligned} \tag{2}$$

[2] The most general form of the problem extends to $\mathbf{z}_i \in \mathbb{R}^m$ with $m > 1$ as in Doostmohammadian, Aghasi, Vrakopoulou and Charalambous (2022) and Lakshmanan and De Farias (2008). For the sake of simplicity and clarity of the problem formulation, here we assume $m = 1$.

with $\underline{x}_i := a_i\underline{z}_i$, $\overline{x}_i := a_i\overline{z}_i$ for $a_i > 0$ and reversed otherwise. The form for the DRA problem presented in (2) is the main focus of our paper. The formulation in (2) involves the minimization of a specific objective function, such as cost or loss, while ensuring a balance in resource demand through the equality constraint $\sum_{i=1}^n x_i = b$. In simpler terms, the optimal solution necessitates that the total resources, represented by $\sum_{i=1}^n x_i$, align precisely with the overall demand $b$. The box constraints $\underline{x}_i \leq x_i \leq \overline{x}_i$ imply that the states are upper and lower bounded. This is typically addressed by the barrier functions or penalty terms in the objective function. In other words, one can approximate these local constraints by additive penalty terms to local objective functions. Some example penalty functions are given in the literature (Bertsekas, 1975; Doostmohammadian, Aghasi, Vrakopoulou et al., 2022; Nesterov, 1998; Wu, Magnusson, & Johansson, 2021). One well-known case of penalty function is,

$$f_i^\sigma = c([z_i - \overline{z}_i]^+ + [\underline{z}_i - z_i]^+), \tag{3}$$

with $[u]^+ = \max\{u, 0\}^\sigma$, $\sigma \in \mathbb{N}$, and $c \in \mathbb{R}^+$. One can show that for sufficiently large $c$ the solutions of the penalized problem can get arbitrarily close to the solution of the problem (2). The penalizing terms (or barrier functions) are generally non-quadratic, see examples in Doostmohammadian, Aghasi, Vrakopoulou et al. (2022) and Wu et al. (2021).

Among the properties of the DRA solution, feasibility is a main concern. Mathematically, we define the set $\mathcal{S}_b = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{x}^\top \mathbf{1}_n = b\}$ for problem (2) (or $\widetilde{\mathcal{S}}_b = \{\mathbf{z} \in \mathbb{R}^n | \mathbf{z}^\top \mathbf{a} = b\}$ for problem (1)) as the feasible set and $\mathbf{x} \in \mathcal{S}_b$ (or $\mathbf{z} \in \widetilde{\mathcal{S}}_b$) as the feasible solution. A feasible solution guarantees that the resource-demand constraint (the coupling constraint) is satisfied and there is no gap between the assigned resources and the demand. If a solution is all-time feasible, it is ensured that at any stopping time of the optimization algorithm, the constraints hold. Note that, under the local box constraints, feasible initialization algorithms are needed and are given, for example, in Cherukuri and Cortés (2015). The feasibility property is more discussed in Section 5.1.

## 3. Preliminaries

### 3.1. Lipschitz continuity, smoothness, and convexity for distributed optimization

Different properties of the objective (or cost) function affect the distributed solution proposed in the literature. The first one is on the smoothness of the objective. The objective function $f : \mathbb{R} \mapsto \mathbb{R}$ is called Lipschitz continuous if there is $K_f \in \mathbb{R}$ such that for any $x_1, x_2 \in \mathbb{R}$,

$$|f(x_1) - f(x_2)| \leq K_f |x_1 - x_2|. \tag{4}$$

The smoothness follows the Lipschitz continuity of the function derivative. If the function is Lipschitz and smooth, typical derivative-based (or gradient-based) convergence analysis can be adopted. Otherwise, the convergence analysis is based on non-smooth models (Cortes, 2008; Doostmohammadian, Aghasi, Vrakopoulou et al., 2022).

For a convex function $f : \mathbb{R} \mapsto \mathbb{R}$ and for all $x_1, x_2 \in \mathbb{R}$ and $\kappa \in (0, 1)$, it holds that

$$f(\kappa x_1 + (1-\kappa)x_2) \leq \kappa f(x_1) + (1-\kappa) f(x_2). \tag{5}$$

where the strict inequality holds for strictly convex functions. For a smooth strictly convex function one can show that $\partial^2 f(x) > 0$ for $x \in \mathbb{R}$ (Bertsekas, Nedic, & Ozdaglar, 2001).

For convex objective functions, it is proved that the optimal solution $\mathbf{x}^* = [x_1^*; \dots; x_n^*]$ to problem (2) satisfies the following (Bertsekas et al., 2001; Xiao & Boyd, 2006),

$$\nabla F(\mathbf{x}^*) = \varphi^* \mathbf{1}_n, \tag{6}$$

with $\mathbf{1}_n$ representing the vector of all 1s, $\nabla F(\mathbf{x}^*) = [\partial f_1(\mathbf{x}_1^*); \dots; \partial f_n(\mathbf{x}_n^*)] \in \mathbb{R}^n$, and $\varphi^* \in \mathbb{R}$. This directly follows from the well-known Karush-Kahn-Tucker (KKT) condition and the method of Lagrange multipliers. Similarly, it can be proved that for the solution to the original problem (1) we have,

$$\nabla \widetilde{F}(\mathbf{z}^*) = \widetilde{\varphi}^* \mathbf{a}. \tag{7}$$

Given that the objective function in (1) and (2) is strictly convex, the optimal solution is unique. This is proved by another relevant concept known as the level set analysis. For $F(\mathbf{x}) : \mathbb{R}^n \mapsto \mathbb{R}$, one can define the level set $L_\gamma(F)$ for $\gamma \in \mathbb{R}$ is the set $L_\gamma(F) = \{\mathbf{x} \in \mathbb{R}^n | F(\mathbf{x}) \leq \gamma\}$. Let assume $F(\mathbf{x})$ is convex; then, all its level sets $L_\gamma(F)$ are also convex $\forall \gamma > 0$ (Bertsekas et al., 2001). For two distinct points $\mathbf{x}_1$ and $\mathbf{x}_2$, let $F(\mathbf{x}_1) > F(\mathbf{x}_2)$. For two associated level sets $L_{\gamma_1}(F)$ and $L_{\gamma_2}(F)$ with $\gamma_1 = F(\mathbf{x}_1), \gamma_2 = F(\mathbf{x}_2)$ one can prove that Iouditski (2015),

$$\nabla F(\mathbf{x}_1)^\top (\mathbf{x}_2 - \mathbf{x}_1) \geq 0. \tag{8}$$

For strictly convex function $F$, the strict inequality in (8) holds. It can be proved that for every feasible set $\mathcal{S}_\mathbf{b}$ and with a strictly convex function there is a unique solution $\mathbf{x}^* \in \mathcal{S}_\mathbf{b}$ that satisfies Eq. (6) (similar statement holds for $\mathbf{z}^* \in \widetilde{\mathcal{S}}_b$ and Eq. (7)) (Doostmohammadian, Aghasi, Vrakopoulou et al., 2022). This is because the level sets of $F(\mathbf{x})$ are strictly convex and implies that only one of its level sets say $L_\gamma(\widetilde{F})$, is adjacent to the constraint facet $\mathcal{S}_\mathbf{b}$, where the touching only happens at one point, say $\mathbf{x}^*$. Since the level set is adjacent to the constraint at $\mathbf{x}^*$, the gradient $\nabla F(\mathbf{x}^*)$ is orthogonal to the constraint facet $\mathcal{S}_\mathbf{b}$ and follows that

$$\nabla F(\mathbf{x}^*) \in \mathrm{span}(\mathbf{1}_n). \tag{9}$$

Using contradiction, consider $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$ in $\mathcal{S}_\mathbf{b}$ satisfying $\nabla F(\mathbf{x}_1^*) = \alpha_1 \mathbf{1}_n$ and $\nabla F(\mathbf{x}_2^*) = \alpha_2 \mathbf{1}_n$. This can be interpreted in two ways: (i) we have two points for which $L_\gamma(F)$, $\gamma = F(\mathbf{x}_1^*) = F(\mathbf{x}_2^*)$ touches the constraint $\mathcal{S}_\mathbf{b}$ which is an affine set. This contradicts the strict convexity of the level set $L_\gamma(F)$; (ii) the points are on the border of two level sets $L_{\gamma_1}(F)$, $\gamma_1 = F(\mathbf{x}_1^*)$ and $L_{\gamma_2}(F)$, $\gamma_2 = F(\mathbf{x}_2^*)$, which are both adjacent to the same affine set $\mathcal{S}_\mathbf{b}$. Now, assume $F(\mathbf{x}_2^*) > F(\mathbf{x}_1^*)$, and following (8) we have,

$$\nabla F(\mathbf{x}_2^*)^\top (\mathbf{x}_1^* - \mathbf{x}_2^*) > 0. \tag{10}$$

However, since both $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$ belong to $\mathcal{S}_\mathbf{b}$ we have $\mathbf{1}_n^\top(\mathbf{x}_1^* - \mathbf{x}_2^*) = 0$ with the level sets both adjacent to $\mathcal{S}_\mathbf{b}$ at both points. This implies that $\nabla F(\mathbf{x}_2^*)^\top (\mathbf{x}_1^* - \mathbf{x}_2^*) = 0$, which contradicts (10). Since both of the mentioned cases (i) and (ii) contradict the properties of the strictly convex level sets, this implies the uniqueness of $\mathbf{x}^* \in \mathcal{S}_\mathbf{b}$ satisfying Eq. (6) for strictly convex objective functions.

It is typically assumed that the local objective functions $f_i(x_i) : \mathbb{R} \mapsto \mathbb{R}$, $i \in \{1, \dots, n\}$ satisfy $\partial^2 f_i(x_i) > 0$ in primal-based formulations and in dual-based formulations only convex assumption is made (see Nedić & Liu, 2018; Yang, Yi et al., 2019 and references therein).[3] Note that the penalty term $f_i^\sigma$ (for every node $i$) is Lipschitz continuous and smooth for $\sigma \in \mathbb{N}_{\geq 2}$ and non-smooth for $\sigma = 1$. Therefore, in primal-based formulations (as in the LG-based solution) it is common to consider $\sigma \in \mathbb{N}_{\geq 2}$.

Different assumptions on the convexity and smoothness of the objective function are discussed in the literature. LG-based solutions mainly assume smooth and strictly (or strongly) convex functions. In the case of non-smooth but locally Lipschitz objective functions, the generalized gradient in the following form is used instead (Cortes, 2008; Doostmohammadian, Aghasi, Vrakopoulou et al., 2022),

$$\partial f(x) = \mathrm{co}\{\lim \nabla f(x_i) : x_i \to x, x_i \notin \Omega_f \cup S\}, \tag{11}$$

with co as the convex hull, $S \subset \mathbb{R}$ as any set of zero Lebesgue measure, and $\Omega_f \in \mathbb{R}$ denoting the non-differentiable set of points. If $f$ is locally Lipschitz at $x$, then $\partial f(\mathbf{x})$ is nonempty, compact, and convex, and the set-valued map $\partial f : \mathbb{R} \to \mathcal{B}\{\mathbb{R}\}$ (with $\mathcal{B}\{\mathbb{R}\}$ as all subsets of $\mathbb{R}$), $\mathbf{x} \mapsto \partial f(\mathbf{x})$, is upper semi-continuous and locally bounded (Cortes, 2008).

[3] Primal and dual formulations are later discussed in details in Section 4.

### 3.2. Algebraic graph theory

The network of nodes/agents[4] (or the multi-agent system) is typically modelled by a (directed) graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where the set of links (or edges) $\mathcal{E}$ denotes the interaction of different nodes/units $\mathcal{V} = \{1, \ldots, n\}$. A directed link $(i, j) \in \mathcal{E}$ implies data transmission from node $i$ to node $j$. This also defines the neighbouring set of nodes, i.e., $i$ is a neighbour of $j$. In general, two neighbouring sets of nodes are defined, the set of in-neighbours $\mathcal{N}_j^- = \{i|(i, j) \in \mathcal{E}\}$ and the set of out-neighbours $\mathcal{N}_j^+ = \{i|(j, i) \in \mathcal{E}\}$. The graph is associated with a weight matrix $W$ (also referred to as the adjacency matrix). Each link $(j, i) \in \mathcal{E}$ is associated with the weight $W_{ij} > 0$ and the matrix $W := [W_{ij}] \in \mathbb{R}_{\geq 0}^{n\times n}$. The zero-nonzero pattern of $W$ follows the topology of $\mathcal{G}$. Define the associated Laplacian matrix $L = [L_{ij}]$ as,

$$L_{ij} = \begin{cases} \sum_{j\in\mathcal{N}_i^-} W_{ij}, & \text{for } i = j, \\ -W_{ij}, & \text{for } i \neq j. \end{cases} \tag{12}$$

Following the definition, $\mathbf{1}_n^\top L = \mathbf{0}_n$ and $L$ has (at least) one zero eigenvalue. More eigen-spectrum properties of the Laplacian matrix can be found in Godsil and Royle (2001) and Olfati-Saber, Fax, and Murray (2007).

There are various assumptions on graph connectivity in the literature. The network could be undirected (bidirectional links), directed and balanced, time-varying (switching), or uniformly connected (B-connected). Existing works in the literature also consider unbalanced digraphs, e.g., see Zhang, You, and Cai (2020). Each case is defined below.

- **Undirected and Symmetric:** Every link in $\mathcal{G}$ is bidirectional with symmetric weight matrix, i.e., $W_{ij} = W_{ji}$. This implies a symmetric weight matrix $W$ and symmetric Laplacian $L$.
- **Balanced Digraph**[5]**:** The graph is possibly directed with the link weights satisfying

  $$\mathbf{1}_n^\top W = W\mathbf{1}_n, \tag{13}$$

  or

  $$\sum_{j\in\mathcal{N}_i^-} W_{ij} = \sum_{j\in\mathcal{N}_i^+} W_{ji}, \ \forall i \in \mathcal{V}. \tag{14}$$

- **Stochastic:** A network is weight-stochastic (or row-stochastic) if

  $$W\mathbf{1}_n = \mathbf{1}_n, \tag{15}$$

  and is bi-stochastic (or doubly stochastic) if it is balanced and,

  $$W\mathbf{1}_n = W^\top\mathbf{1}_n = \mathbf{1}_n. \tag{16}$$
- **Switching:** In case $\mathcal{G}(t) = \{\mathcal{V}, \mathcal{E}(t)\}$ is time-varying (with the same node set $\mathcal{V}$) or the link weights change over time the network is called of switching topology.
- **Uniformly-Connected:** For a time-varying network if the union graph $\mathcal{G}_B(t) = \{\mathcal{V}, \mathcal{E}_B(t)\}$ over a time-window $B > 0$ is connected it is called B-connected or uniformly-connected. In other words, there is $B > 0$ such that the (edge) union graph (defined as follows) is connected for $t \geq 0$.

$$\mathcal{E}_B(t) = \bigcup_t^{t+B} \mathcal{E}(t), \ \ \mathcal{G}_B(t) = \bigcup_t^{t+B} \mathcal{G}(t). \tag{17}$$

The network connectivity depends, for example, on the broadcasting power levels and the communication range of the agents or their mobility over time. The uniform-connectivity is the weakest form of network connectivity over time. This implies $\mathcal{G}(t)$ may lose connectivity at some times while possibly regaining connectivity at some other times. In other words, the connectivity holds over longer time intervals in the case of communication links arbitrarily connecting and disconnecting over the dynamic network. This is the weakest requirement in distributed coordination and consensus literature (Nedić & Olshevsky, 2014; Ren & Beard, 2005) however it is strong enough to guarantee convergence. Different network-connectivity assumptions are given in the literature. The works (Falsone, Notarnicola, Notarstefano, & Prandini, 2020; Rikos et al., 2021) assume weight-stochastic condition while all-time connectivity assumption is considered in Banjac, Rey, Goulart, and Lygeros (2019), Cherukuri and Cortés (2015), Xiao and Boyd (2006) and Wu et al. (2021). Recall that for switching and stochastic networks it is needed to re-adjust (update) the link weights to ensure weight-stochasticity at all times, see Vaidya, Hadjicostis, and Dominguez-Garcia (2012a) for such algorithms.

It is known that the convergence rate of the optimization and scheduling algorithms is tied with the eigen-spectrum of the Laplacian matrix $L$. Among the eigenvalues, the largest eigenvalue $\lambda_n$ and smallest nonzero eigenvalue $\lambda_2$ play a key role in the convergence rate and the optimization step rate. This is a result of Courant–Fischer theorem and consensus algorithms (Olfati-Saber & Murray, 2004). For example, for a balanced Laplacian matrix $L$ and state vector $\mathbf{x} \in \mathbb{R}^n$ and $\overline{\mathbf{x}} =: \mathbf{x} - \frac{\mathbf{1}_n^\top \mathbf{x}}{n}\mathbf{1}_n$ (known as dispersion state vector), a common (disagreement) Lyapunov function is used as follows:

$$\mathbf{x}^\top L_s \mathbf{x} = \overline{\mathbf{x}}^\top L_s \overline{\mathbf{x}}, \tag{18}$$

$$\lambda_2 \|\overline{\mathbf{x}}\|_2^2 \leq \mathbf{x}^\top L_s \mathbf{x} \leq \lambda_n \|\overline{\mathbf{x}}\|_2^2, \tag{19}$$

with $L_s = \frac{L+L^\top}{2}$. Note that for (undirected) connected graphs with weights equal to 1, $\lambda_2$ is called the Fiedler value or the Algebraic connectivity (Gross & Yellen, 2004; Olfati-Saber & Murray, 2004) and is representative of the connectivity of the associated network. For uniformly-connected network $\mathcal{G}_B$, $\lambda_n \leq \lambda_{nB}$ and $\lambda_2 \leq \lambda_{2B}$. This implies that link addition may increase the algebraic connectivity (Olfati-Saber & Murray, 2004; Wang & Van Mieghem, 2008). Therefore, given $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2$, the algebraic connectivity implies that $\lambda_2(\mathcal{G}) \geq \lambda_2(\mathcal{G}_1), \lambda_2(\mathcal{G}) \geq \lambda_2(\mathcal{G}_2)$. Therefore, for uniform-connectivity, we have $\lambda_2(\mathcal{G}_B(t)) \geq \lambda_2(\mathcal{G}(t))$. One way to prove this is by the fact that $\lambda_2(\mathcal{G}) \geq \frac{1}{nd_g}$ (with $d_g$ as the network diameter) (Gross & Yellen, 2004, p. 571). Note that the eigen-spectrum of the network is key in the convergence rate (Olfati-Saber et al., 2007) and the upper bound on the step-rate of the optimization algorithms (Doostmohammadian, Aghasi, Rikos et al., 2022).

### 3.3. Consensus and coordination algorithms

In control theory, consensus in distributed systems implies achieving agreement (or consensus) on a certain quantity/property among a group of interconnected agents or nodes. The goal of consensus algorithms are to synchronize the states or behaviours of these agents so that they reach a common decision or converge to a shared value over a network (Olfati-Saber et al., 2007; Ren & Beard, 2005; Ren, Beard, & Atkins, 2005).

The intuition of a typical consensus algorithm is the following. Each agent represents a node or component that interacts with its neighbours by exchanging information. Consensus algorithms enable agents to communicate and update their states based on the information received from neighbouring agents. The iterative process continues until a consensus or agreement is reached, typically by adjusting the states of the agents to minimize the differences between their values.

[4] Throughout this paper the terms nodes or agents, links or edges, and network or graph are used interchangeably.

[5] There exist distributed algorithms in the literature to ensure weight-balance or weight-stochasticity over the directed networks, see for example Gharesifard and Cortés (2012) and Hadjicostis, Domínguez-García, Charalambous, et al. (2018).

Consensus algorithms achieve convergence when the states of all agents become sufficiently close/equal to each other. Convergence is typically ensured by carefully choosing the weights and the communication topology among the agents (Olfati-Saber et al., 2007). Different weight assignment strategies, such as average consensus (Olfati-Saber et al., 2007; Ren & Beard, 2005), consensus with leader (Cao, Zhang, & Ren, 2015; Tanner, Pappas, & Kumar, 2004), or consensus with time-varying weights (Xiao & Wang, 2008), can be used based on specific system requirements and objectives.

Consensus algorithms find applications in networked control systems (Kia et al., 2019), e.g., coverage control (Bullo, Cortés, & Martinez, 2009; Cortés, Martınez, Karatas, & Bullo, 2004; Doostmohammadian, Sayyaadi, & Moarref, 2009; Sayyaadi & Moarref, 2011; Telsang, 2022), opinion dynamics (Doostmohammadian & Khan, 2014; Mirtabatabaei & Bullo, 2012), rendezvous algorithms (Cortés, Martínez, & Bullo, 2006), flocking and formation algorithms (Olfati-Saber & Murray, 2002; Tanner, Jadbabaie, & Pappas, 2007), target tracking (Doostmohammadian, Taghieh and Zarrabi, 2021; Olfati-Saber & Jalalkamali, 2011), FDI (Doostmohammadian & Meskin, 2020; Doostmohammadian, Zarrabi and Charalambous, 2022), attack detection (Deghat, Ugrinovskii, Shames, & Langbort, 2019; Giraldo et al., 2018), and estimation (Doostmohammadian & Khan, 2013; Kar & Moura, 2013; Park & Martins, 2012). More recent applications are in machine learning and optimization (Doostmohammadian, Aghasi, Charalambous and Khan, 2021; Xin, Kar, & Khan, 2020). These algorithms enable agents to coordinate their actions, reach agreements, and collectively solve problems in a distributed manner. In general, consensus algorithms are crucial for achieving cooperation, coordination, and synchronization among agents in distributed systems.

There are various consensus algorithms used in control theory, with the most well-known being the consensus protocol based on linear consensus dynamics (Olfati-Saber et al., 2007; Ren & Beard, 2005; Sundaram & Hadjicostis, 2008). In this protocol, the agents update their states by incorporating a weighted average of their own state and the states of their neighbours, and the weights determine the influence of each neighbour on the agent's state update. The state update in discrete time is the following:

$$\mathbf{x}(k+1) = W\mathbf{x}(k), \tag{20}$$

with $\mathbf{x}(k)$ being the states of the agents at time step $k$, and $W$ being a stochastic matrix (or more accurately stochastic-indecomposable-aperiodic (SIA) (Ren & Beard, 2005)). In continuous-time, the state update can be expressed as:

$$\dot{\mathbf{x}}(t) = -L\mathbf{x}(t), \tag{21}$$

with $L$ being the Laplacian matrix. It is known that the convergence rate of these consensus protocols and other consensus-based distributed algorithms tightly depends on the algebraic graph properties and spectrum analysis of the associated stochastic matrix $W$ or the Laplacian matrix $L$ (discussed in Section 3.2). In terms of convergence scheme, other than the asymptotic convergence in linear models, finite-time (Bhat & Bernstein, 2000; Doostmohammadian, 2020; Feng & Hu, 2017; Hu & Yang, 2018; Lin, Ren, & Farrell, 2017; Sayyaadi & Doostmohammadian, 2011), fixed-time (Firouzbahrami & Nobakhti, 2022a; Garg & Panagou, 2021; Ning, Han, & Zuo, 2017; Shang, 2017), and prescribed-time (Ji, Yu, Zhang, Guo, Li, 2023; Wang, Song, Hill and Krstic, 2018) models are proposed to guarantee convergence over a finite bounded time-interval in contrast to asymptotic convergence. These algorithms mainly leverage non-Lipschitz sign-based dynamics to accelerate the agreement and convergence.

## 4. Distributed resource allocation algorithms

The existing literature on distributed optimization includes both *constrained* and *unconstrained* scenarios. These two scenarios are directly related as described in Zhang et al. (2020) (and discussed later in Section 4.1.2). The literature on distributed optimization can be classified in terms of being general or problem-specific based on the nature of the optimization problem being addressed. Most works in the literature assume ideal linear models to describe the data exchange among agents in the network and consider no problem-specific (nonlinear) constraints on the agents' actuation dynamics. There are also some works in the literature that address problem-specific constraints in their proposed solutions and other kinds of constraints in their models. In the following subsections, we discuss and analyse both kinds of distributed optimization algorithms.

### 4.1. General algorithms

In the context of distributed optimization, linear models are based on techniques like consensus optimization. These include both unconstrained problems (Khan et al., 2020; Qureshi, Xin, Kar, & Khan, 2020), and problems subject to state constraints (Doan & Beck, 2017; Gharesifard & Cortés, 2013; Jiang & Charalambous, 2021; Wang, Wang, Chen and Wang, 2018). These two are related via the mirror relation between distributed optimization and dual formulation of the DRA described in Section 4.1.2. Other approaches in the literature include, diffusion-based (Alghunaim, Yuan, & Sayed, 2019), event-triggered (Kia, Cortés, & Martínez, 2015), primal–dual (Turan, Uribe, Wai, & Alizadeh, 2021; Uribe, Lee, Gasnikov, & Nedic, 2020; Yazdandoost Hamedani & Aybat, 2017), Lagrangian-based (Aybat & Yazdandoost Hamedani, 2016a; Doan & Beck, 2017), exact ratio consensus (Jiang & Charalambous, 2021), initialization-free (Chen & Guo, 2020; Li, Yi and Xie, 2020; Yi, Hong, & Liu, 2016), gradient push-sum (Yang et al., 2013), push–pull (or AB type algorithm) (Xin, Xi and Khan, 2019), attack-resilient (Kuwaranancharoen et al., 2020; Turan et al., 2021), switching agents (Esteki & Kia, 2022), sign-based (Doostmohammadian & Aghasi, 2024; Shi, Su, Mu, & Sun, 2024), and momentum-based (or accelerated) solutions (Ghadimi, Johansson, & Shames, 2011; Ochoa, Poveda, Uribe, & Quijano, 2019). The solutions and their convergence analysis are given in both continuous-time (Gharesifard & Cortés, 2013; Kia et al., 2015) and discrete-time dynamics (Qureshi et al., 2020; Uribe et al., 2020; Xin, Xi et al., 2019). Other than the *resource allocation problem* in which the coupling constraint is on resource-demand balance, some works in the literature address the distributed optimization subject to consensus-constraint (Alghunaim et al., 2019; Doostmohammadian, Jiang, Liaquat, Aghasi, & Zarrabi, 2024; Jiang & Charalambous, 2021; Wang, Wang et al., 2018). On the other hand, coupling constraints on the sum of resources are considered in the DRA literature, e.g., see the *linear* preliminary works in Doan and Beck (2017), Doan and Olshevsky (2017), Gharesifard and Cortés (2013), Turan et al. (2021) and Nedić, Olshevsky, and Shi (2018). However, note that these works assume no extra constraints on the agents' dynamics such as nonlinearities or time-delays in the communication links and/or agents' actuation. The existing general DRA algorithms are categorized into primal-based, dual-based, and ADMM-based solutions.

#### 4.1.1. Laplacian gradient solutions to the primal-formulation

Laplacian gradient (LG) solutions apply the concept of Laplacian matrix $L$ to coordinate and optimize the resource allocation across multiple agents or nodes in a network. Each agent $i$ updates its local variables based on the gradients of the primal objective functions (its own and its neighbours). The local variable update is given by:

$$x_i(k+1) = x_i(k) - \alpha \sum_{j \in \mathcal{N}_i^-} W_{ij}(\partial_{x_i} f_i(k) - \partial_{x_j} f_j(k)), \tag{22}$$

with $\partial_{x_i} f_i$ as the derivative of $f_i$ with respect to $x_i$, and $\alpha$ as the step-size. In case the function is nonsmooth, the generalized gradient is used instead in Eq. (22). With a slight abuse of notation, one can rewrite the above solution in compact LG form:

$$\mathbf{x}(k+1) = \mathbf{x}(k) - \alpha L \nabla F(k), \tag{23}$$

where the bound on step-size $\alpha$ to ensure convergence can be defined based on algebraic properties of the associated Laplacian matrix (the eigen-spectrum properties in Section 3.2), the convexity properties of the objective function; see Doostmohammadian, Aghasi, Rikos et al. (2022) and Xiao and Boyd (2006) for more details. The equivalent continuous-time version of (23) can be written in the following form:

$$\dot{\mathbf{x}} = -\alpha L \nabla F. \tag{24}$$

For strictly convex objective functions and initializing from a state that satisfies resource-demand constraint (a feasible initialization) there is only one unique optimal convergence point to the above dynamics that matches the solution of the standard DRA problem (2) in the form (6). This is proved by level set analysis in Section 3.1 (see more details in Doostmohammadian, Aghasi, Pirani et al., 2022; Doostmohammadian, Aghasi, Vrakopoulou et al., 2022). The main feature of this approach is that, for feasible initialization, resource-demand constraint $\sum_{i=1}^n x_i = b$ holds at all times and all iterations of the solution dynamics. This is because we have

$$\mathbf{1}_n^\top \mathbf{x}(k+1) = \mathbf{1}_n^\top \mathbf{x}(k) - \alpha \mathbf{1}_n^\top L \nabla F(k) = \mathbf{1}_n^\top \mathbf{x}(k), \tag{25}$$

where the last equality follows from the fact that $\mathbf{1}_n^\top$ is the left eigenvector associated with the zero eigenvalue of the Laplacian matrix $L$. This is referred to as all-time feasibility (or anytime feasibility) and is discussed later in Section 5.1. For the given dynamics (22) or (24) one can see that the equilibrium point $\mathbf{x}^*$ is only in the form $\nabla F(\mathbf{x}^*) \in \text{span}(\mathbf{1}_n)$ for which $\dot{\mathbf{x}} = \mathbf{0}$ or $\mathbf{x}(k+1) = \mathbf{x}(k)$. One can prove by contradiction that there is no other equilibrium for the dynamics (22) or (24) with $\nabla F(\mathbf{x}^*) \notin \text{span}(\mathbf{1}_n)$. Suppose by contradiction that the equilibrium point $\mathbf{x}^*$ satisfies $\partial_{x_i} f_i(\mathbf{x}_i^*) \neq \partial_{x_j} f_j(\mathbf{x}_j^*)$ for (at least) two nodes $i, j$. Let $\alpha = \text{argmax}_{i\in\{1,\dots,n\}} \partial_{x_i} f_i(\mathbf{x}_i^*)$ and $\beta = \text{argmin}_{i\in\{1,\dots,n\}} \partial_{x_i} f_i(\mathbf{x}_i^*)$ over the multi-agent system. Then, over the network of agents $\mathcal{G}(t)$ (or its uniformly connected union $\mathcal{G}_B(t)$) there is (at least) one path between agents $\alpha$ and $\beta$ or vice versa. Over such a path, there exists two agents $\overline{\alpha}$ and $\overline{\beta}$ for which $\partial_{x_i} f_{\overline{\alpha}}(\mathbf{x}_{\overline{\alpha}}^*) \geq \partial_{x_i} f_{\mathcal{N}_{\overline{\alpha}}}(\mathbf{x}_{\mathcal{N}_{\overline{\alpha}}}^*), \partial_{x_i} f_{\overline{\beta}}(\mathbf{x}_{\overline{\beta}}^*) \leq \partial_{x_i} f_{\mathcal{N}_{\overline{\beta}}}(\mathbf{x}_{\mathcal{N}_{\overline{\beta}}}^*)$ with $\mathcal{N}_{\overline{\alpha}}$ and $\mathcal{N}_{\overline{\beta}}$ respectively as the set of neighbours of $\overline{\alpha}$ and $\overline{\beta}$ with the strict inequality for (at least) one neighbour in $\mathcal{N}_{\overline{\alpha}}$ and $\mathcal{N}_{\overline{\beta}}$. Therefore, if the network is always connected or uniformly connected over time we get $\dot{\mathbf{x}}_{\overline{\alpha}}^* < 0$ and $\dot{\mathbf{x}}_{\overline{\beta}}^* > 0$ or in discrete-time $\mathbf{x}_{\overline{\alpha}}^*(k+1) < \mathbf{x}_{\overline{\alpha}}^*(k)$ and $\mathbf{x}_{\overline{\beta}}^*(k+1) > \mathbf{x}_{\overline{\beta}}^*(k)$. These contradict the definition of equilibrium for $\mathbf{x}^*$ and imply that for the LG-type dynamics in the form (22) or (24) one can easily prove that the equilibrium $\nabla F(\mathbf{x}^*) \in \text{span}(\mathbf{1}_n)$ over uniformly-connected multi-agent networks.

*Accelerated LG* primal-based solutions accelerate the convergence rate of the LG-based dynamics by adding momentum (Ghadimi et al., 2011; Ochoa et al., 2019) or by adding nonlinear non-Lipschitz sign-based functions (Doostmohammadian & Aghasi, 2024; Doostmohammadian, Aghasi, Pirani et al., 2022; Garg, Baranwal, Hero, & Panagou, 2019; Parsegov, Polyakov, & Shcherbakov, 2013) and improve the convergence rate. In the latter case, the sign-based functions are widely used in finite/fixed-time dynamical systems to reach faster convergence, see Feng and Hu (2017), Firouzbahrami and Nobakhti (2022a), Hu and Yang (2018), Lin et al. (2017) and Ning et al. (2017) for example. However, this makes the protocol nonsmooth and, therefore, the analysis must follow nonsmooth (discontinuous) system dynamics analysis as in Cortes (2006, 2008).

#### 4.1.2. Dual-formulation gradient tracking solutions

Dual-problem-based gradient-tracking (GT) solutions use the concept of dual variables and gradient tracking to satisfy the equality constraints while optimizing the objective function (Zhang et al., 2020). Here we explain dual-problem-based GT solutions for equality-constraint DRA. The existing methods exploit the relationship between distributed optimization and DRA (Alghunaim, Lyu, Yan, & Sayed, 2021; Li & Su, 2022; Nedić et al., 2018; Wu, Liu, & Zhu, 2022; Xu et al., 2017; Xu, Zhu, Soh, & Xie, 2018; Zhang et al., 2020; Zhu, Ren, Yu, & Wen, 2019) and the idea of consensus-based small-gain theorem (Zholbaryssov, Hadjicostis, & Dominguez-Garcia, 2022). The dual problem is formulated by introducing dual variables corresponding to the equality constraints. Let $\lambda_i$ be the dual variable associated with the (more general) equality constraint $A_i x_i = b_i$. The dual problem aims to maximize the dual function with respect to the dual variables $\lambda_i$ and is defined as:

$$\max_{\lambda_i} \Phi(\lambda) = \min_x L_\rho(x, \lambda), \tag{26}$$

where $L(x, \lambda)$ is the Lagrangian function defined as:

$$L(x, \lambda) = F(x) + \sum_{i=1}^n \lambda_i^\top (A_i x_i - b_i). \tag{27}$$

The key idea for solving (26) is to track the gradients of the Lagrangian function with respect to the dual variables. One solution is that each agent $i$ updates its dual variable based on the gradients obtained from the Lagrangian function as:

$$\lambda_i(k+1) = \lambda_i(k) + \alpha(k) \nabla_{\lambda_i} L(x(k), \lambda(k)), \tag{28}$$

with $\alpha(k)$ as the step-size (or the learning rate) at iteration $k$. After updating the dual variables, each agent $i$ solves its local optimization problem to update the local variables $x_i$ while keeping the dual variables fixed. This can be done using standard optimization techniques to minimize the local objective function subject to the local equality constraint. A consensus step is then performed to improve agreement and convergence among the local variables (Alghunaim et al., 2021; Zhang et al., 2020). Specifically, the agents exchange information and update their local variables to improve consensus. This step ensures that the local variable solutions satisfy the global equality constraints. The algorithm continues executing with these three steps until a convergence criterion is met, which may rely on various criteria such as the change in the objective function, the primal and dual residuals, or the number of iterations. There are also primal–dual-based DRA dynamics to optimize *non-smooth* objective functions (Huang, Meng, Sun, & Ren, 2023; Huang, Meng, Sun, & Wang, 2024).

#### 4.1.3. Dual-formulation ADMM-based solutions

The Alternating-Direction-Method-of-Multipliers (ADMM) is an optimization technique that is particularly useful for solving problems with separable objective functions or constraints. It is an iterative method that combines the advantages of decomposition, proximal methods, and dual decomposition. This method is shown to be very effective for solving large-scale distributed optimization and learning models (Boyd et al., 2011; Iutzeler, Bianchi, Ciblat, & Hachem, 2015). Classic ADMM methods are based on a semi-centralized method, known as Parallel-ADMM (Bertsekas & Tsitsiklis, 2015). Recently, many variations of the distributed version of this method are proposed in the literature (Aybat & Yazdandoost Hamedani, 2019; Chang, 2016; Doostmohammadian, Jiang and Charalambous, 2022; Falsone, Margellos, & Prandini, 2018; Jiang, Doostmohammadian, & Charalambous, 2022; Makhdoumi & Ozdaglar, 2017; Wei & Ozdaglar, 2012). There are many works devoted to dual consensus ADMM (Banjac et al., 2019; Chang, 2016; Chang, Hong, & Wang, 2014; Jian, Hu, Wang, & Shi, 2019) where the consensus ADMM cannot be applied on the primal-formulation but it is adopted on the dual formulation. Some other literature is focused on Jacobi-like ADMM (Deng, Lai, Peng, & Yin, 2017), and proximal ADMM (Yang, Jia, Xu, Guan and Spanos, 2022).

Here, we present a common distributed ADMM model for resource allocation (see Yang, Jia et al., 2022 for details). The local constraint at each agent $i$ is denoted as $g_i(x_i)$ and captures the local resource allocation requirements or limitations. In the equality-constraint resource allocation formulation, $g_i(x_i)$ is in the form $A_i x_i = b_i$ with the coupling constraint $\sum_i b_i = b$. The ADMM solution is defined over the optimally dual formulation of the original problem based on augmented Lagrangian formulation. The augmented Lagrangian function for ADMM is defined as follows:

$$L_\rho(x, \lambda) = F(x) + \sum_{i=1}^n \lambda_i^\top (A_i x_i - b_i) + \frac{\rho}{2} \sum_{i=1}^n \|A_i x_i - b_i\|^2. \tag{29}$$

Here, $\lambda_i$ is the dual variable associated with the agent $i$, and $\rho$ is a penalty parameter. To solve the problem, the agents perform iterative updates to optimize the local variables and the dual variables. The steps are as follows:

(i) Local variable update: Each agent $i$ solves its local optimization subproblem by minimizing the augmented Lagrangian function with respect to its local variables $x_i$ while keeping the dual variables fixed. This minimization is defined as:

$$x_i(k+1) = \underset{x_i}{\operatorname{argmin}}\, L_\rho(x, \lambda). \tag{30}$$

This can be achieved using standard optimization techniques, such as gradient-based methods or closed-form solutions depending on the problem structure.

(ii) Dual variable update: Each agent $i$ updates its dual variable $\lambda_i$ based on the current local variable solution and the global dual variables:

$$\lambda_i(k+1) = \lambda_i(k) + \rho(A_i x_i(k+1) - b_i). \tag{31}$$

(iii) Consensus update: A consensus step is performed to achieve agreement and convergence among the agents. The local variable updates and dual variable updates naturally along with the consensus update ensure consistency between the agents. Note that the primal and dual updates of ADMM both violate the coupling constraint and, therefore, the use of dynamic average consensus algorithms to track the feasibility-constraint violation distributedly is adopted.

ADMM iterates until the convergence criterion is met. The key idea behind ADMM is the alternating updates of $x$ variables while maintaining consistency with the dual variable $\lambda$. The algorithm iteratively solves subproblems that are relatively easier to solve than the original problem. The augmented Lagrangian terms help enforce the constraints and balance the primal–dual convergence. ADMM is particularly effective when the objective function and constraints can be decomposed into separable parts. This allows for parallel or distributed computation, as different nodes or processors can independently optimize their local variables $x_i$ and communicate their results to update the dual variable $\lambda$. By iteratively updating the local variables and the dual variables, ADMM enables agents to coordinate their resource allocations in a distributed manner while respecting the constraints imposed by the problem. It is important to note that the specific mathematical expressions and algorithms may vary depending on the problem formulation and the specific constraints and objectives of the resource allocation problem at hand. The formulations described here provide a general framework for understanding how ADMM can be applied to equality-constraint resource allocation. Other than resource allocation, ADMM has been successfully applied in various fields, including machine learning, signal processing, image reconstruction, and network optimization. Compared to the gradient-based approaches, ADMM has both advantages and disadvantages. Specifically, ADMM is well-suited for problems that can be decomposed into smaller, easier-to-solve subproblems. In these cases, compared to gradient-based methods, ADMM can be faster, leading to more efficient parallelization (Boyd et al., 2011).

### 4.2. Problem-specific solutions

Most real-world applications consider nonlinear constraints. Linear models, although useful in some aspects, are limited in their ability to capture complex dynamics and interactions. By studying problem-specific models, we can develop more realistic and accurate representations of the existing distributed resource allocation systems. Nonlinear constraints exist in both node dynamics and data exchange among the agents. For example, in AGC the node dynamics are constrained with the so-called RRLs that represent saturation-type nonlinear models on the rate of change on the node states (Doostmohammadian, Aghasi, Vrakopoulou et al., 2022). On the other hand, quantization (Grammenos, Charalambous, & Kalyvianaki, 2021) and clipping exist in the communication systems that exchange information among the nodes/agents. Additionally, it is important to note that some nonlinear models add useful properties to the system, for example, faster convergence and finite/fixed-time solutions (Doostmohammadian, Aghasi, Pirani et al., 2022). Many existing solutions to distributed algorithms are mainly limited to finite-time (Chen, Ren, & Feng, 2016; Doostmohammadian, 2020) or fixed-time (Doostmohammadian, Aghasi, Pirani et al., 2022; Garg et al., 2019; Ning et al., 2017; Parsegov et al., 2013) convergence models. Such sign-based dynamics are also prevalent in consensus literature (Doostmohammadian, 2020; Stanković, Beko, & Stanković, 2019; Wei, Everts, Camlibel, & van der Schaft, 2017) which also allow for robust and noise-resilient design (Doostmohammadian & Aghasi, 2023). Other works also develop realistic models considering the effect of actuation saturation or clipping in the data channels. Some existing works study data quantization (Hadjicostis, Vaidya, & Domínguez-García, 2016; Nedic, Olshevsky, Ozdaglar, & Tsitsiklis, 2008; Nekouei, Alpcan, Nair and Evans, 2016; Nekouei, Nair and Alpcan, 2016; Rikos et al., 2020; Rikos & Hadjicostis, 2020; Wei, Yi, Sandberg, & Johansson, 2019) and some study saturation and clipping (Liu, Saberi, Stoorvogel and Nojavanzadeh, 2020; Wei et al., 2019). Such nonlinear modellings make distributed (both constrained and unconstrained) optimization challenging in terms of computation, accuracy, feasibility, optimality, and convergence. Other than these works that consider only a specific problem, some works focus on a *general* model to account for all types of model constraints on the agents' dynamics. These works may include, e.g., a combination of the aforementioned nonlinear models (Doostmohammadian, Aghasi, Vrakopoulou et al., 2022; Doostmohammadian, Aghasi and Zarrabi, 2023; Doostmohammadian, Vrakopoulou, Aghasi and Charalambous, 2022). An example problem-specific DRA solution is given below:

$$x_i(k+1) = x_i(k) - \alpha \sum_{j\in\mathcal{N}_i^-} W_{ij} g_n\Big(g_l(\partial_{x_i} f_i(k)) - g_l(\partial_{x_j} f_j(k))\Big), \tag{32}$$

where $g_n(\cdot)$ denotes possible nonlinearity on the node dynamics and $g_l(\cdot)$ denotes possible nonlinearity on the links. A composition of nonlinearities is considered in the above model. For converging to the optimal value, these nonlinearities must follow some specific properties (Doostmohammadian, Aghasi, Vrakopoulou et al., 2022; Doostmohammadian, Aghasi, Pirani et al., 2022. The convergence is proved under sign-preserving odd nonlinear mappings for both $g_n(\cdot)$ and $g_l(\cdot)$. An example is when the nonlinear mapping follows upper and lower sector-bound conditions, for example, logarithmic quantization (Doostmohammadian, Aghasi, Rikos et al., 2022). The feasibility of the solution holds for undirected networks (Doostmohammadian, Aghasi, Vrakopoulou et al., 2022; Doostmohammadian, Vrakopoulou et al., 2022). In case, there is no node nonlinearity (i.e., $g_n(x) = x$) the solution converges to the optimal allocation over weight-balanced networks (Doostmohammadian & Aghasi, 2024; Doostmohammadian, Aghasi, Pirani et al., 2022; Doostmohammadian, Aghasi, Rikos et al., 2022). These nonlinear solutions might be designed to address data quantization, to handle saturation/clipping, to converge in predefined time, and to address resiliency and robustness to noise. Note that it is not easy to address these nonlinearities with the existing dual-formulation-based solutions, e.g., the ADMM method.

### 4.3. Machine-learning-based solutions

Other than traditional methods which rely on predefined heuristics or optimization techniques, the advent of machine learning (ML) has opened new avenues for developing adaptive algorithms, particularly in dynamic and uncertain environments. ML approaches for distributed resource allocation use historical data to learn patterns and make predictions about resource demands, usage, and optimal allocation strategies. In the following, we include some of these techniques.

Supervised learning models, trained on labelled datasets, predict resource allocation requirements based on input features (Amiri & Mohammad-Khanli, 2017; Morariu, Morariu, Răileanu, & Borangiu, 2020). For example, regression models can forecast the future demand for resources based on historical consumption patterns (Amiri & Mohammad-Khanli, 2017; Zhang, Cherkasova, & Smirni, 2007). In this regard, many distributed supervised learning algorithms can be adopted to train and classify the data and learn specific patterns (Boyd et al., 2011; Doostmohammadian, Aghasi, Zarrabi, 2023; Nedic, 2020; Sundhar Ram, Nedić, & Veeravalli, 2012). Unsupervised learning techniques such as clustering can identify groups of similar users or applications, helping to improve resource allocation strategies (Wang, Eisen, & Ribeiro, 2021); for example, using K-means clustering to segment users with similar resource needs (Kumar, Chauhan, & Yadav, 2021; Liu, Jia and Ding, 2020). Multi-agent reinforcement learning (RL) is particularly useful for scenarios where multiple agents must collaboratively or competitively allocate resources. In this case, each agent learns to make decisions based on its own objective and the actions of other agents, leading to the emergence of cooperative resource allocation strategies through exploration and exploitation processes, for example, resource allocation/scheduling over IoT networks (Gu, Zhang, Lin, & Alazab, 2020; Hussain, Hussain, Anpalagan, & Benslimane, 2020), distributed data centres (Wei et al., 2022), device-to-device (D2D) communications (Jayakumar & Nandakumar, 2023), and UAV-aided multi-access edge computing (MEC) systems (Nie, Zhao, Gao, & Yu, 2021). In contrast to the centralized learning approaches which gather resource/demand data from all agents, distributed RL allows agents to hold individual policies based on local observations. Deep reinforcement learning, on the other hand, combines deep learning with RL to process complex input data and learn high-dimensional policies, allowing for resource allocation decisions in large and *dynamic* environments, for example in resource scheduling for space-air-ground integrated networks (Zhang et al., 2022), Vehicle-to-vehicle (V2V) communication (Gündoğan, Gürsu, Pauli, & Kellerer, 2020), and spectrum/power allocation over networks (Nasir & Guo, 2019; Yang, Zhao et al., 2022). In distributed setups, where agents are interconnected, *Graph Neural Networks (GNNs)* can model the relationships and dependencies between agents' objectives to facilitate informed resource allocation. By considering the graph structure formed by agents and their interactions, GNNs help in learning policies that account for inter-agent allocation dynamics (Gu, She, Quan, Qiu, & Xu, 2023; Wang, Eisen and Ribeiro, 2022). In decentralized settings where data privacy is a concern, federated learning allows agents to collaboratively learn a shared objective model while keeping their local data private (Guo et al., 2022; Ji & Qin, 2022; Ji, Qin and Tao, 2023). This is especially relevant in scenarios like IoT (Nguyen, Sharma, Vu, Chatzinotas, & Ottersten, 2020), where resource allocation decisions need to be made with limited data sharing. These ML approaches, particularly those utilizing RL in multi-agent settings, offer promising directions for improving distributed resource allocation strategies (in uncertain setups) and adapting to dynamic objectives, see Lim et al. (2021) for instance.

### *4.4. Convergence rate*

The convergence rate refers to the speed (number of iterations) at which a distributed optimization algorithm reaches a state that is sufficiently close to the optimal solution. Fast convergence is important in various applications, e.g., in time-critical applications where a quick response is required, or in scenarios with limited computational resources, as it reduces the overall computation time and resource consumption. For example, fast convergence is crucial in EDP and AGC over the power grid since the optimization problem needs to be terminated over finite time intervals (e.g., the applications in Sections 6.1 and 6.3). Moreover, the convergence rate impacts the scalability of the DRA algorithms as they are commonly employed over large-scale systems.

The convergence rate of existing distributed optimization problems and DRA may differ based on the adopted techniques. Most DRA literature claims linear convergence for their algorithms. In this case, the residual (i.e., the difference between the current state and the optimal state) decreases geometrically (linear in log scale) with respect to the number of iterations (Xin & Khan, 2018). In most existing works, linear convergence is considered to be desirable and indicates a sufficiently-fast convergence rate, e.g., see Chang et al. (2014), Doan and Olshevsky (2017), Haddadpour, Kamani, Mahdavi, and Cadambe (2019), Ling and Ribeiro (2013), Nedic, Olshevsky, and Shi (2017), Nedić et al. (2018), Nedić, Olshevsky, Shi, and Uribe (2017), Olshevsky (2015), Scaman, Bach, Bubeck, Lee, and Massoulié (2017) and Stich (2019). Some of the existing algorithms establish linear convergence for particular quadratic-form objective functions; for example, consensus-based scenarios for CPU scheduling (Rikos et al., 2021) or economic dispatch (Kar & Hug, 2012). However, in most cases, the objective is non-quadratic, e.g., because of extra penalty terms to address the box constraints. For non-quadratic objectives, some ADMM-based solutions prove an ergodic rate of convergence for both the residual and feasibility violation (Aybat & Yazdandoost Hamedani, 2016a, 2016b). Some optimization papers propose algorithms with superlinear (faster than linear) convergence rates (Beck, Nedić, Ozdaglar, & Teboulle, 2014; Zargham, Ribeiro, Ozdaglar, & Jadbabaie, 2013). Additionally, different algorithms and techniques for unconstrained distributed optimization can be used to improve the convergence rate, such as acceleration methods (Ghadimi et al., 2011; Scaman et al., 2017; Xin, Sahu, Khan and Kar, 2019), adaptive learning rates (Chen & Sayed, 2012; Chen, Tu, & Sayed, 2011; Dauphin, De Vries, & Bengio, 2015; Liu et al., 2019), or variance reduction techniques (Ghadikolaei & Magnússon, 2020; Li, Cen, Chen and Chi, 2020; Qureshi, Xin, Kar, & Khan, 2021; Xin et al., 2020).

In general, for problems subject to feasibility constraints the asymptotic convergence should be fast enough to result in near-optimal solutions before the termination time of the algorithm. For all-time feasible but asymptotic solutions (i.e., the algorithms in Section 4.1.1), as the algorithm evolves over time the solution gets closer to the optimal value while satisfying the feasibility constraint at all times. On the other hand, for dual-based and ADMM solutions (i.e., the algorithms in Sections 4.1.2 and 4.1.3) the convergence is ensured to be fast enough to reach all-time feasibility asymptotically as fast as possible. In general, dual-based and ADMM solutions are known to be faster than primal-based solutions in terms of number of iterations for convergence, however, they are computationally more complex and each iteration is more time-consuming. Some works in the literature propose strategies that converge in finite-time (Chen et al., 2016; Doostmohammadian, Aghasi, Vrakopoulou et al., 2022; Li, Lin, Cai and Yan, 2020; Wang, Fei, & Wu, 2020), fixed-time (Chen & Guo, 2020; Dai, Fang, & Jia, 2022; Doostmohammadian & Aghasi, 2024; Doostmohammadian, Aghasi, Pirani et al., 2022; Li & Ding, 2020; Shi, Xu, Yang, Lin, & Wang, 2022; Song, Cao, & Rutkowski, 2021), or prescribed-time (predefined-time) (Gong, Cui, Shen, Xiong, & Huang, 2022; Guo & Chen, 2022; Lin, Wang, Li, & Yu, 2020b). Recall that finite-time convergence implies reaching the optimal state in finite time interval depending on the initial values, while fixed-time convergence does not depend on the initial states. The prescribed-time algorithms, on the other hand, converge in a predefined time interval irrespective of the initial states. These algorithms are mainly based on non-Lipschitz sign-based mapping, and in discrete time, may result in unwanted oscillations (*chattering* phenomena) around the equilibrium.

### *4.5. Summary of key points*

Here, we summarize some key points discussed in this section. In general, dual-based GT and ADMM solutions converge faster than primal-based LG solutions. However, some LG algorithms use (i) momentum terms and (ii) nonlinear sign-based mapping to accelerate the

**Table 1**
Comparison between existing DRA algorithms.

| Algorithm | Formulation | Rate | Smoothness |
|---|---|---|---|
| LG | Primal | Slow | Smooth |
| Accelerated LG | Primal | Fast | Nonsmooth |
| GT | Dual | Average | Smooth |
| ADMM | Dual | Fast | Smooth |

rate of convergence. Recall that, for the latter case, the sign nonlinearity makes the solution nonsmooth, which mandates convergence analysis based on discontinuous dynamical systems. Table 1 summarizes these comparisons based on different properties, including formulation type, convergence rate, and smoothness of the solution.

## 5. Main features and properties

In this section, we discuss different key features of the existing DRA solutions in terms of their assumptions on feasibility constraints, network connectivity, delay-tolerance, and packet-drops.

### 5.1. Anytime feasibility

In DRA anytime feasibility implies that resource-demand balance holds during all iterations. In other words, as the optimization algorithm evolves in time, the coupling constraint on the states is always satisfied (Cherukuri & Cortés, 2015; Wu et al., 2021). In real-world applications, by maintaining the resource-demand balance, the system can better meet the quality of service (QoS) guarantees, ensuring that the allocated resources are sufficient to satisfy the desired demand levels. This is of importance in, for example, economic dispatch in power networks or automatic generation control. In this application, this ensures the produced power meets the power load demand at all times. Violating the resource-demand balance may cause service disruption, power delivery issues, and even breakdown in the power grid (Cherukuri & Cortés, 2015; Wu et al., 2021). So it is crucial to preserve this balance at all times. Ensuring feasibility in network structures often relies on the crucial feature of weight-balancing. The majority of LG-based solutions presume the presence of weight-balanced digraphs or undirected graphs to maintain feasibility (Cherukuri & Cortés, 2015; Doostmohammadian, Aghasi, Rikos et al., 2022).

As detailed in Section 4.1.1, LG-based solutions hold all-time resource-demand feasibility (the coupling-constraint) for feasible initialization. This indicates that at any termination time of these algorithms, the resource-demand feasibility holds and there would be no constraint violation (Cherukuri & Cortés, 2015; Doostmohammadian & Aghasi, 2024; Doostmohammadian, Aghasi, Pirani et al., 2022; Doostmohammadian, Aghasi, Rikos et al., 2022; Doostmohammadian, Aghasi, Vrakopoulou et al., 2022). In contrast, the feasibility in most optimal dual-formulation-based solutions (such as in Sections 4.1.2 and 4.1.3) is asymptotic and, therefore, the convergence must be fast enough to reach (almost) feasibility within the algorithm running time (Aybat & Yazdandoost Hamedani, 2016a; Nedić et al., 2018; Wang, Hong, Sun, & Liu, 2019). In general, anytime feasibility does not hold in the ADMM-based literature, but some works claim to reach feasibility sufficiently fast during the run-time of the algorithm (Banjac et al., 2019; Falsone et al., 2020). Dynamic average consensus has been used to track the feasibility violation in a distributed way (Li & Su, 2022). Feasibility-violation tracking is studied in dual-formulation-based gradient-tracking over unbalanced network (Zhang et al., 2020) and in dual-based ADMM-tracking (Kia, 2017). Note that for all-time feasibility, the agents need to take initial values which are feasible, i.e., in the beginning, the sum of resources must meet the demand. There are some algorithms in the literature to set the feasible initialization of states, for example, see the algorithm in Cherukuri and Cortés (2015). On the other hand, initialization-free solutions are proposed in the literature (Chen & Guo, 2020; Ji, Yu et al., 2023; Lin, Wang, Li, & Yu, 2021; Yi et al., 2016); these works do not need specific feasible initialization to converge, but they converge to the feasible solution over time.

### 5.2. Network reliability

In a distributed optimization setting, multiple nodes or entities collaborate to solve a complex optimization problem by sharing information, exchanging messages, and coordinating their actions over a network. The network acts as the communication infrastructure that enables these interactions. In distributed optimization algorithms, the network is typically modelled as an undirected or directed graph with nodes representing the agents and links as their interactions (message-passing or communication). The properties of the network infrastructure are modelled over the graph topology; for example, packet drop is modelled as link failure (Doostmohammadian, Khan and Aghasi, 2022) or failed agents as node removal (Li, Ho, & Lu, 2016). Additionally, in some optimization literature, the knowledge of the network topology is local (Sundaram & Gharesifard, 2016), i.e., each agent only knows its neighbouring topology and not the entire network.

Distributed optimization algorithms often assume the presence of all nodes throughout the optimization process. However, in real-world scenarios, nodes can fail due to hardware failures, software crashes, or network partitions. When a node fails, it disrupts communication and coordination among other nodes, potentially impacting the overall optimization performance. Robustness to node failures becomes crucial to ensure the algorithm can continue making progress even in the presence of such failures (Chen & Sayed, 2012; Chen et al., 2011). On the other hand, some works study optimization under the arrival/addition of new nodes (Falsone, Margellos, Zizzo, Prandini, & Garatti, 2022). Some works are devoted to studying optimality and convergence in the presence of deception attacks (Fu, Ma, Qin, & Kang, 2021; Shao, Wang, Wang, & Liu, 2020), denial-of-service (DoS) attacks (Sardana & Joshi, 2009), byzantine man-in-the-middle attacks (Turan, Uribe, Wai, & Alizadeh, 2020; Wang, Liu and Ling, 2022; Xu & Liu, 2022; Xu, Liu, & Huang, 2022), and malicious agents (Yemini, Nedić, Gil, & Goldsmith, 2022). In these works the network links or the nodes could be corrupted and under attack. On the other hand, some works propose privacy-preserving strategies for optimal DRA to avoid some of these attacks (Angel, Kannan, & Ratliff, 2020; Beaude, Benchimol, Gaubert, Jacquot, & Oudjane, 2020; Chen & Li, 2021; Jacquot, Beaude, Benchimol, Gaubert, & Oudjane, 2019; Lin, Wang, Li, & Yu, 2020a). In these works the agents avoid to reveal their private information on the constraints, gradients, and individual solution profile to a third party. Subsequent subsections will delve into the examination of network reliability concerns, encompassing connectivity, latency, and packet drops.

#### 5.2.1. Networks subject to connectivity considerations

Distributed optimization algorithms typically rely on iterative processes to reach an optimal solution. The convergence speed depends on how quickly nodes can exchange information and update their local models or variables. If the network experiences frequent disruptions or high latency, the convergence speed may be significantly affected, leading to slower optimization and potentially suboptimal results. Note that in many applications, the network might be dynamic (time-varying or switching). This is, for example, due to agents' mobility or change in their broadcasting range. At some iterations, the network may even get disconnected and regain connectivity over time. Therefore, in many works the convergence is proved over uniformly-connected networks (Doostmohammadian, Aghasi, Pirani et al., 2022; Doostmohammadian, Aghasi, Rikos et al., 2022; Doostmohammadian, Aghasi, Vrakopoulou et al., 2022; Nedic & Ozdaglar, 2009), in contrast to all-time connectivity in many other works (Rikos et al., 2021; Wu et al., 2021; Xin & Khan, 2018; Xin, Sahu et al., 2019). Uniform network connectivity is common in mobile sensor network applications, where some links may connect and disconnect as the mobile sensors (agents or robots) move into and out of broadcast range of one another. The connectivity might be lost at some times due to link failures while preserving uniform connectivity over some finite time intervals. See the mathematical modelling in Section 3.2 for details. In the existing literature, most dual-formulation-based and ADMM-based solutions in Sections 4.1.2 and 4.1.3 assume all-time network connectivity. On the other hand, LG-based solutions defined on the primal formulation in Section 4.1.1 prove convergence over uniformly-connected networks (Doostmohammadian, 2023; Doostmohammadian & Aghasi, 2024; Doostmohammadian, Aghasi, Pirani et al., 2022; Doostmohammadian, Aghasi, Rikos et al., 2022; Doostmohammadian, Aghasi, Vrakopoulou et al., 2022, 2023), which is a privilege of these works.

*5.2.2. Networks subject to time-delays*

In a distributed setting, nodes often need to exchange information, such as optimization variables, gradients, or updates. If the network experiences delays, it can significantly impact the communication between nodes. Increased communication delays can hinder the convergence of the optimization algorithm and slow down the overall coordination process. The effect of communication delays in coordination and consensus literature is vastly studied, see for example (Doostmohammadian, Khan, Pirani and Charalambous, 2021; Hadjicostis & Charalambous, 2014; Olfati-Saber & Murray, 2004; Seuret, Dimarogonas, & Johansson, 2008) and references therein. One trivial approach to handle time delays is to update over a longer time scale after receiving all delayed information (Hadjicostis & Charalambous, 2014). However, a more challenging and desirable approach is to update at the same time scale via all the delayed and non-delayed received information (Doostmohammadian, Khan et al., 2021; Olfati-Saber & Murray, 2004; Seuret et al., 2008). On the other hand, asynchronous data transmission over the links is considered. This is the case where the receiver and the transmitter clocks are not necessarily synchronized and are independent. Therefore, distributed algorithms are needed to synchronize the clocks over the communication network to prevent issues raised by asynchronous cases (Schenato & Fiorentin, 2011; Schenato & Gamba, 2007). In the distributed optimization setting latency may cause the solution to diverge. Few works address homogeneous delays or asynchronous data exchange (Wang et al., 2019; Zhu et al., 2019), and some other works discuss how latency affects unconstrained or consensus-constraint optimization (Agarwal & Duchi, 2012; Al-Lawati & Draper, 2020; Hadjicostis & Charalambous, 2011; Wang, Wang et al., 2018, 2018). Note that latency may result in feasibility gap (violation of the coupling-constraint) (Wang et al., 2019) while LG-based delay-tolerant solutions lead to no feasibility gap over networks, see Chen and Zhao (2017) and Doostmohammadian, Aghasi, Rikos et al. (2022).

Typically, addressing communication time delays is more convenient in LG-type solutions. Consider a network of agents where each agent's available information consists of its own data without delay and all data possibly delayed from its neighbours up to that point. Generally, it is assumed that the delays are upper bounded, for example by $\bar{\tau}$ steps, to prevent packet drops over the links. The communicated packets over the multi-agent network are time-stamped and every node $i$ knows the time at which agent $j \in \mathcal{N}_i^-$ sent its data over the link $(j,i)$. To make the mathematical analysis more convenient, define an indicator function $\mathcal{I}_{k,ij}$ capturing the delay $\tau_{ij}(k) \leq \bar{\tau}$ on the link $(j,i)$,

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

**Table 2**
Summarizing the main properties of existing DRA algorithms.

| Algorithm | Feasibility | Convexity | Connectivity | Delay/Packet-Drop |
|---|---|---|---|---|
| LG | All-Time | Strict | Uniform | Resilient |
| GT | Asymptotic | Regular | All-Time | Prone |
| ADMM | Asymptotic | Regular | All-Time | Prone |

Then, one can rewrite the delay-tolerant version of the LG-type dynamics to solve resource allocation as,

$$x_i(k+1) = x_i(k) - \alpha \sum_{j\in\mathcal{N}_i^-} \sum_{r=0}^{\bar{\tau}} W_{ij}(\partial_{x_i} f_i(k-r) - \partial_{x_j} f_j(k-r))\mathcal{I}_{k-r,ij}(r). \tag{34}$$

Note that to avoid any feasibility gap, the weighted difference of the gradients over every link $(j,i)$ is considered. For all-time feasibility some more assumptions on the delay are needed: the network should be undirected and $\tau_{ij}(k) = \tau_{ji}(k)$. The delays at different links could be arbitrary, heterogeneous, and time-varying in general. Convergence over a nonlinear setup can be established for the following delay-tolerant dynamics (as demonstrated in Doostmohammadian, Aghasi, Rikos et al. (2022))

$$\begin{aligned} x_i(k+1) =& x_i(k) \\ &- \alpha \sum_{j\in\mathcal{N}_i^-} \sum_{r=0}^{\bar{\tau}} W_{ij}\Big(g_l(\partial_{x_i} f_i(k-r)) - g_l(\partial_{x_j} f_j(k-r))\Big)\mathcal{I}_{k-r,ij}(r). \end{aligned} \tag{35}$$

*5.2.3. Networks subject to packet drops*

DRA algorithms should be designed to handle large-scale systems with a high number of nodes. However, as the network scales up, maintaining reliable communication between all nodes becomes more challenging. Network reliability issues, such as a high probability of packet loss, can become more prevalent in large-scale distributed systems, potentially impacting the performance and scalability of the optimization and coordination algorithm. In this direction, some works address coordination and convergence in the presence of packet drops in the distributed setups, see for example robust consensus under packet drops (Fagnani & Zampieri, 2009; Hadjicostis, Vaidya, & Domínguez-García, 2015; Liu, You, Yang, & Zhao, 2015; Vaidya, Hadjicostis, & Domínguez-García, 2012b). In DRA setup, convergence under packet drops cannot be easily addressed by the existing ADMM solutions, and therefore, ideal communications are considered, for example, see Banjac et al. (2019) and Falsone et al. (2020). One reason is the feasibility violation. In the LG-type setup packet drops are addressed based on link removal, see for example Doostmohammadian, Khan et al. (2022).

*5.3. Summary of key points*

Here, we summarize some key points discussed in this section. In general, the primal-based LG algorithms preserve all-time constraint feasibility, require strict convexity of the objective function, can be designed resilient to (possible) delay and packet-drops, and converge over uniformly connected networks that may change over time and lose/regain connectivity. On the other hand, dual-based GT and ADMM algorithms reach constraint feasibility asymptotically, only require convexity (and not strict convexity) of the objective, require all-time connectivity for convergence, and are generally prone to delay and packet-drops. These are summarized in Table 2.

## 6. Applications of DRA

In this section, we provide motivating examples and real-world applications of DRA.

### 6.1. Distributed energy resource management over smart grids

Distributed energy resource management refers to the optimization of power generation and allocation across multiple distributed energy resources in an electrical grid. Traditionally, this is a centralized process where a central authority determines the optimal allocation of power generation resources to meet the load demand while minimizing costs (Chowdhury & Rahman, 1990; Xia & Elaiw, 2010). However, in a distributed setting, where there are numerous small-scale generation sources like solar panels, wind turbines, batteries, and microgrids, the traditional centralized economic dispatch suffers from the drawbacks of the centralized solutions and may not be suitable on large scale. By enabling individual energy units or small clusters of energy units to autonomously optimize their power generation and consumption decisions, one can decentralize this process. In the distributed approach, each unit or cluster assesses the local conditions, such as energy prices, local demand, and availability of renewable energy sources, to make optimal decisions regarding their power generation and consumption (Chen & Zhao, 2017; Cherukuri, Martínez, & Cortés, 2014; Doostmohammadian, Vrakopoulou et al., 2022; Elsayed & El-Saadany, 2014; Firouzbahrami & Nobakhti, 2022b; Pourbabak, Luo, Chen, & Su, 2017). By coordinating and optimizing these distributed decisions, the overall system efficiency and cost-effectiveness can be improved. The solution often relies on advanced control and communication technologies to facilitate information exchange and coordination among the units (Doostmohammadian, 2023). It can incorporate real-time data, predictive models, and optimization algorithms to dynamically adjust power generation and consumption in response to changing conditions.

In the context of distributed systems, this problem focuses on managing a diverse set of resources such as distributed generation (e.g., solar, wind, and biomass), energy storage (e.g., batteries), electric vehicles, demand response programs, and smart grid technologies. The goal is to *coordinate* and optimize the operation of these resources to meet energy demands, improve system reliability, and achieve economic and environmental objectives. It involves tasks such as forecasting energy generation and consumption, scheduling and dispatching resources, managing energy storage, coordinating charging and discharging, optimizing demand response programs, and integrating renewable energy sources into the grid. Advanced analytics, optimization algorithms, and control systems are often utilized to enable efficient decision-making and resource coordination. Other factors like grid constraints, environmental considerations, market dynamics, and regulatory policies ensure the optimal utilization of energy resources while maintaining grid stability and meeting sustainability goals (Boqiang & Chuanwen, 2009).

**Mathematical Model:** The distributed optimization problem for EDP is formulated as minimizing the sum of local energy cost functions (Doostmohammadian, 2023; Doostmohammadian, Aghasi, Vrakopoulou et al., 2023) (subject to the equality-constraint mentioned later):

$$\min_{\mathbf{x},\mathbf{y}} \quad F(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{m} f_i(x_i) + \sum_{j=m+1}^{n} e_j(y_j), \tag{36}$$

with $x_i, y_j \in \mathbb{R}$ as the power state of the generator node $i \in \{1,\dots,m\}$ and battery node $j \in \{m+1,\dots,n\}$. Following the formulation (1) vector $\mathbf{z} = [x_1;\dots;x_m;y_{m+1};\dots;y_n]$ denotes the global state, $\widetilde{f}_i(z_i) = f_i(x_i)$ as the local cost of generators and $\widetilde{f}_j(z_j) = e_j(y_j)$ as the local cost of battery nodes. The feasibility constraint coupling the energy resources and the demand is the following equality-constraint:

$$\sum_{i=1}^{m} x_i = b_{dem} + \sum_{j=m+1}^{n} y_j. \tag{37}$$

This implies that the summation of the produced powers equals the demand $b_{dem} \in \mathbb{R}$ plus the summation of the reserved powers. This *energy resource-demand feasibility* constraint is of crucial importance and needs to hold at all times; otherwise, the generated and reserved power balance is violated which may cause service disruption and even system break-down. Further, the local box constraints on the energy nodes imply the limiting range of produced/reserved powers at generators/batteries,

$$\underline{x}_i \leq x_i \leq \overline{x}_i, \ \ \underline{y}_j \leq y_j \leq \overline{y}_j. \tag{38}$$

It should be recalled from KKT condition in Section 3, from the formulation (1) and the optimal state (7), one can replace $\mathbf{a} = [\mathbf{1}_m, -\mathbf{1}_{n-m}]$ in the optimal gradient formulation in (7).

### 6.2. Distributed CPU scheduling and task allocation over data centers

Distributed CPU scheduling and task allocation over data centres involve the efficient allocation of computational tasks across multiple CPUs or computing nodes/servers in a distributed computing environment, see Fig. 2.

In such a system, there are multiple CPUs or computing nodes that can execute computational tasks (Grammenos, Charalambous, & Kalyvianaki, 2023; Rikos, Nylöf, Gracy, & Johansson, 2022). Distributed CPU scheduling aims to optimize the allocation of tasks to these CPUs in order to achieve efficient resource utilization, load balancing, and overall system performance. Traditional centralized CPU scheduling algorithms, such as those used in standalone operating systems, are not directly applicable in distributed environments due to the distributed nature of resources and the need for inter-node communication. In distributed CPU scheduling, the task allocation decisions are typically made in a distributed manner, where individual computing nodes collaborate to make scheduling decisions based on local information.

Distributed CPU scheduling algorithms often consider factors such as CPU load, resource availability, network latency, and task characteristics (e.g., priority, estimated execution time) to determine the most suitable node for task execution. The goal is to evenly distribute the workload among the available CPUs, minimize task response time, and optimize resource utilization (Chen et al., 2022; Kalyvianaki, Charalambous, & Hand, 2009; Makridis, Deliparaschos, Kalyvianaki, Zolotas, & Charalambous, 2020; Vlahakis et al., 2021). On the other hand, task allocation over data centres further refers to the assignment of computational tasks to specific data centres or cloud resources (see an example in Hattab et al., 2019). In a large-scale distributed computing environment, data centres are geographically distributed and may have different computing capacities, network connectivity, and service-level agreements. Task allocation aims to optimize the utilization of these resources while satisfying performance objectives and resource constraints (Mann, 2015).

**Mathematical Model:** The data centre is modelled as a set of $n$ computing servers each can also operate as a resource scheduler (which is a standard procedure in modern data centres). Define the set $\mathcal{J}$ as all tasks to be scheduled and $b_j \in \mathcal{J}$ as the task which requires $\rho_j$ cycles to be executed. Additionally, define $T_h$ as the time that optimization runs the tasks on the servers. The CPU capacity during the optimization operation is then equal to $\pi_i^{\max} := c_i T_h$, with parameter $c_i$ denoting the sum of all clock rate frequencies of all processing cores at node $i$ (Rikos et al., 2021). Let the total demanded resources be $\rho := \sum_{b_j \in \mathcal{J}} \rho_j$. The cost formulation for this problem is then in standard distributed optimization form (2) with the quadratic cost defined as (Grammenos et al., 2023; Rikos et al., 2021),

$$\min_{\mathbf{x}} \quad F(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i) = \sum_{i=1}^{n} \frac{1}{2\pi_i^{\max}} (x_i - \rho_i)^2, \tag{39}$$

with $x_i$ as the workload to be assigned to node/server $i$ and the workload-demand constraint (or resource-demand feasibility) as $\sum_{i=1}^{n} x_i = \rho$ and some box constraints make the solution non-trivial, in general. These box constraints follow the fact that we need to keep the operating point of the servers away from the overall capacity; in fact, below $70-80\%$ of full capacity (because of the uncertainty of the processing times) since the mean response time of the servers grows

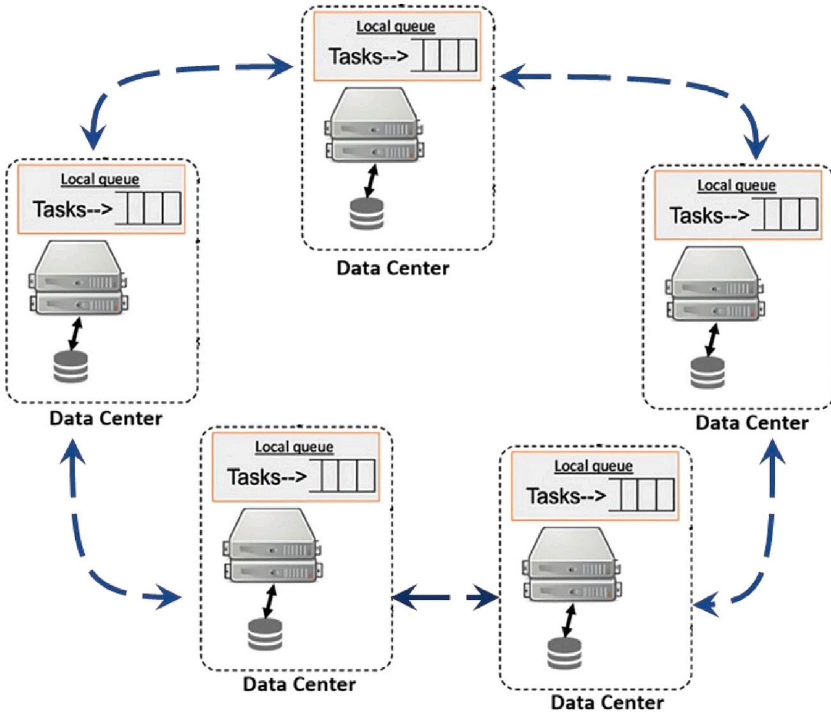


**Fig. 2.** Distributing the task allocation among a group of networked data centres: every server is associated with a cost function and the algorithm optimally assigns the computing tasks to the servers only based on local information exchange.

(exponentially) at some point (Makridis et al., 2020). This concern is addressed by box constraints on the load-to-capacity ratios in the form $0 \leq \frac{x_i+\rho_i}{\pi_i^{\max}} \leq 0.75$ as a rule-of-thumb which is modelled via proper extra penalty terms in the objective function (Doostmohammadian, Aghasi, Rikos et al., 2022).

### 6.3. Distributed automatic generation control

Distributed automatic generation control is a control mechanism used in power systems to maintain the balance between power generation and load demand across multiple control areas or regions. It involves the coordination and control of various generating units distributed throughout the power grid to regulate the system frequency and ensure stable operation. This problem has some similarities with the economic dispatch problem particularly from an optimization viewpoint and the nature of the optimization variables and cost functions (Boqiang & Chuanwen, 2009; Li, Zhao, & Chen, 2015). In a power grid, the electricity demand is continuously changing, and the generation must be adjusted in real time to match the load. Note that, as compared to the economic dispatch problem in Section 6.1, automatic generation control pertains to the real-time adjustment of the generation outputs of power plants to maintain the balance between supply and demand, ensuring system stability and frequency control. It adjusts the generation outputs based on the deviations between scheduled and actual system conditions. While economic dispatch deals with the *long-term* planning of generation resources, automatic generation control addresses the *short-term* fluctuations and variations in demand and generation, working to maintain the stability and reliability of the power grid in real time.

Automatic generation control is traditionally implemented in a centralized manner (Kumar & Kothari, 2005), where a single control centre monitors the system frequency and issues control signals to the generating units to adjust their power output. However, in large interconnected power grids, this centralized control approach may face challenges in terms of communication delays, scalability, and system resilience (Bevrani & Hiyama, 2017; Kumar & Kothari, 2005; Ullah, Basit, Ullah, Aslam, & Herodotou, 2021). Distributed automatic generation control addresses these challenges by decentralizing the control mechanism and allowing individual control areas or regions to autonomously adjust their power generation based on local measurements and information. Each control area has its own processing system, which monitors the local frequency and exchanges information with neighbouring control areas (Bevrani & Hiyama, 2017; Ullah et al., 2021; Zhou et al., 2020) (see Fig. 3).

**Mathematical Model:** Given the power mismatch, e.g., due to some failed generators, the distributed optimization problem is to cooperatively allocate this power mismatch to the rest of the generators while optimizing the costs. The problem is as follows:

$$\min_{\mathbf{x}} \sum_{i=1}^{n} f_i(x_i), \tag{40}$$

$$\text{s.t.} \sum_{i=1}^{n} x_i = P_{mis}, \quad -\underline{x}_i \leq x_i \leq \overline{x}_i$$

where $P_{mis}$ is the power mismatch due to the failed generators, $x_i$ is the state of the power state at generator $i$, $\underline{x}_i, \overline{x}_i$ are the upper and lower limits of generated power at the generator $i$. The coupling constraint $\sum_{i=1}^{n} x_i = P_{mis}$ implies that the mismatch in the power is equal to the new extra allocated powers to the existing generators. The cost function is typically in the quadratic form $f_i(x_i) = \gamma_i x_i^2 + \beta_i x_i + \alpha_i$ with $\gamma_i, \beta_i, \alpha_i$ defined based on the generator type (fuelled by oil, gas, coal, etc.). Some examples of these parameters are given in Table 3 (Wood, Wollenberg, & Sheblé, 2013; Yang et al., 2013).

Moreover, in some literature, some realistic ramp-rate-limit (RRL) constraints are also considered, implying that the rate of change of the generated powers is constrained, i.e.,

$$-R \leq \dot{x}_i \leq R \tag{41}$$

where $\dot{x}_i$ implies the rate of change in the generator power and $R$ denotes the RRL. This problem follows the standard formulation (2) and, therefore, the optimal condition on the states follows (6), i.e., $\mathbf{a} = \mathbf{1}_n$ in (7).

Some existing methods in power generation networks in the AGC setup include Chen et al. (2016), Cherukuri and Cortés (2015), Kar and

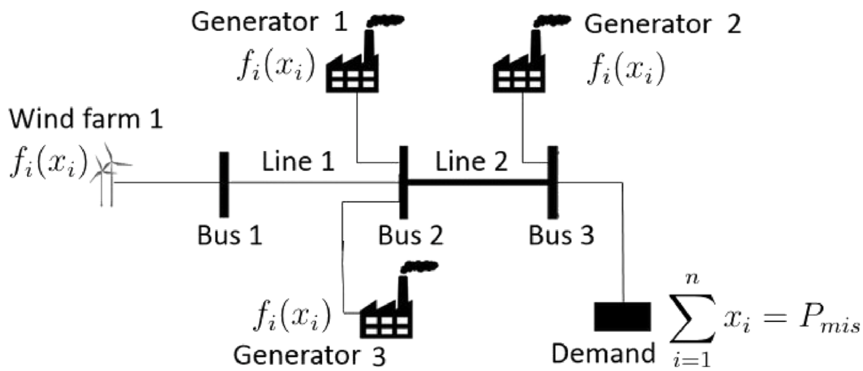


**Fig. 3.** A group of generators producing electricity to meet the load demand: in case there is a power mismatch between the generated power and the demand, the generators readjust their produced power to meet this mismatch while optimizing the sum of local generation cost functions.

**Table 3**
The cost parameters of the generated powers based on the generator types (Wood et al., 2013; Yang et al., 2013).

| Type | $\alpha_i$ (\$/h) | $\beta_i$ (\$/MWh) | $\gamma_i$ (\$/MW$^2$ h) | $\overline{x}_i$ (MW) |
|---|---|---|---|---|
| A | 561 | 2.0 | 0.04 | 80 |
| B | 310 | 3.0 | 0.03 | 90 |
| C | 78 | 4.0 | 0.035 | 70 |
| D | 561 | 4.0 | 0.03 | 70 |
| E | 78 | 2.5 | 0.04 | 80 |

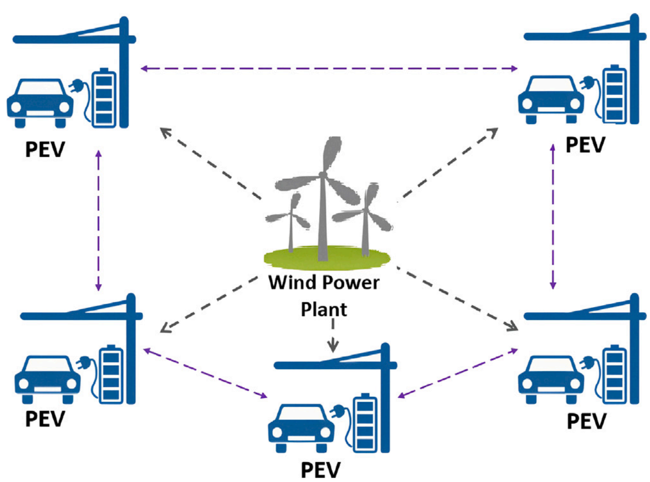


**Fig. 4.** Allocating the charging stations to the fleet of PEVs: each charging station and PEV is associated with an electricity cost and the distributed algorithm optimally assigns the PEVs to the charging stations such that the energy cost is minimum.

Hug (2012), Xiao and Boyd (2006), Yang et al. (2013), Yi et al. (2016) and Zhu et al. (2019). However, these works failed to address the so-called RRLs. More recent solutions address RRLs because of the saturated generator dynamics (Doostmohammadian, Aghasi, Vrakopoulou et al., 2022).

### 6.4. Distributed PEV optimal charging schedule

The distributed PEV (plug-in electric vehicle) optimal charging schedule refers to the decentralized and coordinated scheduling of charging activities for a fleet of electric vehicles, see Fig. 4. The objective is to optimize the charging schedules of individual vehicles while considering factors such as electricity grid constraints, user preferences, and system-level objectives (Sun, Yang, & Yan, 2017). Distributed PEV optimal charging schedule involves individual electric vehicles making charging decisions autonomously based on local information. Instead of a centralized controller dictating the charging schedules for all vehicles, each vehicle independently determines its charging schedule based on its own constraints, preferences, and available information (Kisacikoglu, Erden, & Erdogan, 2017; Li, Gao, Chen, Zhao, & Shen, 2019; Mukherjee & Gupta, 2016). The goal of the distributed PEV optimal charging schedule is typically to minimize the cost, maximize user convenience, or enhance the overall system efficiency. The optimization objective can vary depending on the specific application and stakeholder requirements. For example, the cost objective may aim to minimize the electricity bill for charging, while the user convenience objective may prioritize charging at preferred times or accommodating specific time constraints (Abdalrahman & Zhuang, 2017; Kisacikoglu et al., 2017; Li et al., 2019; Mukherjee & Gupta, 2016). Distributed PEV optimal charging schedule takes into account grid constraints to ensure that charging activities do not overload the electricity grid. By considering factors such as peak demand periods, available grid capacity, and grid stability, the charging schedules can be adjusted to prevent excessive load on the grid and promote grid reliability. Additionally, distributed charging schedules can also incorporate demand response mechanisms, allowing vehicles to respond to dynamic pricing signals or grid signals for load management purposes (Amin et al., 2020; Dallinger & Wietschel, 2012). Various optimization algorithms can be employed by electric vehicles to determine their charging schedules. These algorithms consider the available information, constraints, and objectives to find an optimal or near-optimal charging schedule (Esmaili & Rajabi, 2014; Falsone, Margellos, Garatti, & Prandini, 2017).

**Mathematical Model:** A fleet of $n$ electric vehicles needs to select a schedule to charge their internal batteries via shared charging stations to minimize the electricity costs (Deori, Margellos, & Prandini, 2016; Falsone et al., 2017, 2020; Vujanic, Esfahani, Goulart, Mariéthoz, & Morari, 2016). There exist both vehicle-level constraints, e.g., user preferences on the final state of charge and physical limitations of the batteries, and grid-wide constraints, e.g., maximum deliverable power. In this section, we omit the details of these constraints (refer to Vujanic et al. (2016) for details) and state the main formulation related to the DRA problem (1). Let variable $\mathbf{y}_i$ include the charging and discharging parameters $u_i, v_i \in \{0, 1\}$ and $e_i$ as the charge level of battery $i$. Optimizing whether to charge each vehicle or not to charge, i.e., some states in $\mathbf{y}_i$ follow 0-1 discrete values implying charging or not-charging of the vehicles, implies a mixed integer programming (Ioan, Prodan, Olaru, Stoican, & Niculescu, 2021; Vujanic et al., 2016). Reformulating the problem differently compared to Vujanic et al. (2016), we aim to optimize the charging rate of the batteries (Falsone et al., 2017, 2020; Li, Yi et al., 2020). Then, the distributed optimization problem is in the form,

$$\min_{\mathbf{y}_1,\dots,\mathbf{y}_n} \sum_{i=1}^{n} \gamma_i^\top \mathbf{y}_i, \tag{42}$$

$$\text{s.t.} \sum_{i=1}^{n} A_i \mathbf{y}_i \leq b, \ \mathbf{y}_i \in \mathcal{Y}_i$$

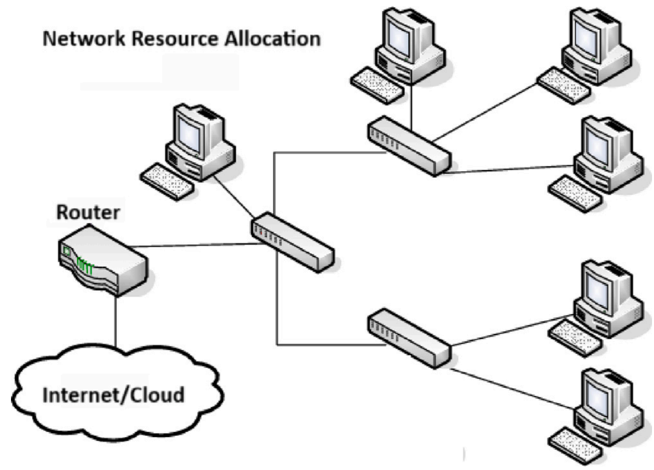


**Fig. 5.** Allocating the network bandwidth, power, or capacity to a group of users: each user/resource pair is associated with a cost and the distributed algorithm optimally assigns the network resources to the users such that the objective utility function is maximized.

with $\mathcal{Y}_i$ as a set determining the local constraint at node $i$.

Note that the coupling constraint in (42) is an inequality constraint. To reformulate the problem in the standard DRA formulation, consider auxiliary or slack variable $s_i$ in the coupling constraint (Falsone et al., 2020). Then, the problem (42) changes to the standard form as

$$\min_{\mathbf{y}_1,\ldots,\mathbf{y}_n} \sum_{i=1}^{n} \gamma_i^\top \mathbf{y}_i, \qquad (43)$$
$$\text{s.t.} \sum_{i=1}^{n} A_i \mathbf{y}_i + s_i = b,\ \mathbf{y}_i \in \mathcal{Y}_i,\ 0 \leq s_i \leq \overline{s}$$

Then, the identification variable is $\mathbf{x}_i = [\mathbf{y}_i; s_i]$ and the cost function is $f_i(\mathbf{x}_i) = \gamma_i^\top \mathbf{y}_i$.

### 6.5. Distributed network utility maximization

Network utility maximization (NUM) is a fundamental concept in the field of network optimization, focusing on the efficient allocation of resources and the maximization of system-wide utility in communication networks (Chiang, Low, Calderbank, & Doyle, 2007). In this framework, the resources, such as bandwidth, power, or capacity, are distributed among network users or components to maximize overall utility or performance, see Fig. 5. Each user or component in the network is associated with a utility function that quantifies their preference for allocated resources. This objective is to maximize the aggregate utility of all users or components in the network while addressing resource constraints such as bandwidth limits, energy constraints, or capacity restrictions. These constraints need to be satisfied while optimizing the objective. This objective is to maximize the aggregate utility of all users or components in the network while addressing the resource constraints.

NUM problems are often complex and high-dimensional, making direct optimization challenging. Decomposition methods are used to break down the problem into smaller, more manageable sub-problems, allowing for efficient optimization (Palomar & Chiang, 2006). NUM plays a critical role in various applications including, telecommunication networks (Lin, Shroff, & Srikant, 2006; Srikant & Ying, 2013), spectrum allocation (Hasan et al., 2016; Tsiropoulos, Dobre, Ahmed, & Baddour, 2014), network energy management (Kraning, Chu, Lavaei, & Boyd, 2013), Internet content delivery (Ercetin & Tassiulas, 2003; Trossen & Parisis, 2012), and Cloud computing (Ergu, Kou, Peng, Shi, & Shi, 2013; Guo, Liu, Yang, Xiao, & Li, 2018). A review of more recent applications in Fog and IoT networks is provided by Chiang and Zhang (2016).

**Mathematical Model:** This problem models a communication network with a set of $\mathcal{L}$ links, where the link/channel has certain finite capacity denoted by $c$. These channels are shared with a set of $n$ sources or nodes denoted by $s$. Each source in $s$ uses a set of $\mathcal{L}(s) \in \mathcal{L}$ of links with certain utility function $U_s(x_s)$ where state $x_s$ denotes the transmission rate of the source. The network utility maximization is modelled as (Chiang et al., 2007; Kelly, Maulloo, & Tan, 1998)

$$\max_{\mathbf{x}_1,\ldots,\mathbf{x}_n} \sum_{s} U_s \mathbf{x}_s, \qquad (44)$$
$$\text{s.t. } R\mathbf{x} \leq c$$

with $R$ as network congestion protocol defined as:

$$R_{ls} = \begin{cases} 1, & \text{if } l \in \mathcal{L}(s) \text{ source } s \text{ uses link } l, \\ 0, & \text{otherwise.} \end{cases} \qquad (45)$$

Similar to (42) this problem can be reformulated to standard DRA formulation as in (43), by use of auxiliary/slack variables in the coupling constraint.

### 6.6. Other applications

There are many other diverse applications of DRA where the formulation is a bit different compared to the above-mentioned cases. Some examples of these applications are listed next where efficient resource allocation and coordination among distributed entities are essential for optimal system performance and resource utilization.

#### 6.6.1. Multi-robot coordination systems for coverage

Distributed resource allocation is crucial in multi-robot systems where multiple robots collaborate to perform tasks in complex environments. It involves allocating tasks and resources, such as sensing capabilities, processing power, and sensor coverage, to different robots to optimize the overall cost/objective, achieve task completion, and ensure coordination among the robots (Bullo et al., 2009; Cortés et al., 2004; Doostmohammadian et al., 2009; Sayyaadi & Moarref, 2011; Telsang, 2022).

#### 6.6.2. Transportation and logistics

In transportation and logistics systems, DRA is essential for optimizing the allocation of vehicles, routes, and delivery tasks. It involves distributing tasks to different vehicles based on their locations, capacities, and availability to minimize delivery time, reduce fuel consumption, and improve overall efficiency (Baldacci et al., 2012; Gao & Niu, 2021; Taillard et al., 1996).

#### 6.6.3. Spreading processes and control of epidemic disease

In the spread of viruses or diseases over a network, different models for epidemic spreading are considered, where the most common model is linearized Susceptible-Infected-Susceptible (SIS). A distributed application over multi-agent social networks is to locally compute the optimal allocation of vaccination resources (for prevention) and antidotes (for treatment) to control the contagion (Enyioha, Jadbabaie, Preciado, & Pappas, 2015; Nowzari, Preciado, & Pappas, 2016).

## 7. Conclusion and future directions

This paper presents a comprehensive review of distributed algorithms for resource allocation over multi-agent systems, including mathematical formulations, relevant applications, various existing solutions, and algorithmic properties. The exploration of these topics has shed light on the challenges and potential solutions in this domain, providing a better comparative analysis of the existing strategies. Specifically, the pros and cons of the primal and dual optimization methods are discussed in detail, giving insight into when/how to apply each method. The comparative analysis in terms of feasibility, convergence, and network properties further enlightens the specific applications and also limitations of each method. Our thorough analysis of the algorithmic properties enables readers to understand the trade-offs involved in employing different distributed algorithms. By comparing convergence rates, communication networking overhead, and constraint feasibility, we provide nuanced insights into the strengths and weaknesses of each approach.

To summarize some of the key points, primal-based DRA algorithms achieve convergence with less stringent requirements and maintain all-time constraint feasibility, although they tend to converge at slower rates compared to dual-based solutions. Primal-based solutions can converge under more relaxed connectivity conditions (uniform connectivity over time), unlike dual-based solutions that demand all-time connectivity. On the other hand, dual-based algorithms only necessitate the convexity of the objective cost function, while primal-based solutions require a stricter assumption of strict/strong convexity. Existing research on primal-based approaches also offers insights into their delay-tolerance and resilience to packet-loss. Moreover, ML and RL-based approaches are adopted in scenarios with dynamic (time-varying) objectives and uncertain setups.

As the field of distributed resource allocation continues to progress, numerous avenues for future research emerge, opening new opportunities for advancements in this domain. Investigating dynamic adaptation in algorithms to accommodate changing system conditions, such as fluctuating resource availability, (mobile) agent arrivals/departures, or evolving demand patterns, presents a promising focus. Areas like asynchronous scheduling (Grammenos et al., 2021), robust and noise-resilient solutions (Stanković et al., 2019; Wei et al., 2017), and low-bit communication-efficient strategies (Doostmohammadian, 2020; Zayyani, Korki, & Marvasti, 2015) stand as examples for future exploration. Developing adaptive resource allocation strategies holds the potential to create more responsive and efficient systems that effectively handle dynamic environments. Furthermore, integrating machine learning and artificial intelligence techniques into resource allocation algorithms could enhance decision-making processes significantly. Exploring reinforcement learning, deep learning, or evolutionary algorithms (Kumar, Kaul, & Hu, 2022; Peteiro-Barral & Guijarro-Berdiñas, 2013) offers an exciting prospect to improve resource allocation efficiency and adaptability. Additionally, the pursuit of data-based modelling and optimization (Mohajer, Barari, & Zarrabi, 2017) holds promise as an interesting research direction. Finally, given the rising emphasis on energy efficiency and sustainability, future research could focus on resource allocation algorithms explicitly incorporating energy consumption (green computing) and environmental impact as optimization criteria.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

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

## References


Abdalrahman, A., & Zhuang, W. (2017). A survey on PEV charging infrastructure: Impact assessment and planning. *Energies*, *10*(10), 1650.

Agarwal, A., & Duchi, J. C. (2012). Distributed delayed stochastic optimization. In *51st IEEE conference on decision and control* (pp. 5451–5452).

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine*, *40*(8), 102–114.

Al-Lawati, H., & Draper, S. C. (2020). Gradient delay analysis in asynchronous distributed optimization. In *IEEE international conference on acoustics, speech and signal processing* (pp. 4207–4211).

Alghunaim, S., Lyu, Q., Yan, M., & Sayed, A. (2021). Dual consensus proximal algorithm for multi-agent sharing problems. *IEEE Transactions on Signal Processing*, *69*, 5568–5579.

Alghunaim, S. A., Yuan, K., & Sayed, A. H. (2019). A proximal diffusion strategy for multiagent optimization with sparse affine constraints. *IEEE Transactions on Automatic Control*, *65*(11), 4554–4567.

Amin, A., Tareen, W., Usman, M., Ali, H., Bari, I., Horan, B., et al. (2020). A review of optimal charging strategy for electric vehicles under dynamic pricing schemes in the distribution charging network. *Sustainability*, *12*(23), 10160.

Amiri, M., & Mohammad-Khanli, L. (2017). Survey on prediction models of applications for resources provisioning in cloud. *Journal of Network and Computer Applications*, *82*, 93–113.

Angel, S., Kannan, S., & Ratliff, Z. (2020). Private resource allocators and their applications. In *IEEE symposium on security and privacy* (pp. 372–391). IEEE.

Aybat, N. S., & Yazdandoost Hamedani, E. (2016a). Distributed primal-dual method for multi-agent sharing problem with conic constraints. In *50th IEEE asilomar conference on signals, systems and computers* (pp. 777–782).

Aybat, N., & Yazdandoost Hamedani, E. (2016b). A primal-dual method for conic constrained distributed optimization problems. *Advances in Neural Information Processing Systems*, *29*.

Aybat, N., & Yazdandoost Hamedani, E. (2019). A distributed ADMM-like method for resource sharing over time-varying networks. *SIAM Journal on Optimization*, *29*(4), 3036–3068.

Baldacci, R., Mingozzi, A., & Roberti, R. (2012). Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, *218*(1), 1–6.

Banjac, G., Rey, F., Goulart, P., & Lygeros, J. (2019). Decentralized resource allocation via dual consensus ADMM. In *IEEE American control conference* (pp. 2789–2794).

Beaude, O., Benchimol, P., Gaubert, S., Jacquot, P., & Oudjane, N. (2020). A privacy-preserving method to optimize distributed resource allocation. *SIAM Journal on Optimization*, *30*(3), 2303–2336.

Beck, A., Nedić, A., Ozdaglar, A., & Teboulle, M. (2014). An $O(1/k)$ gradient method for network resource allocation problems. *IEEE Transactions on Control of Network Systems*, *1*(1), 64–73.

Bertsekas, D. P. (1975). Necessary and sufficient conditions for a penalty method to be exact. *Mathematical Programming*, *9*(1), 87–99.

Bertsekas, D. P., Nedic, A., & Ozdaglar, A. E. (2001). *Lecture notes*, *Convexity, duality, and lagrange multipliers*. MIT Press.

Bertsekas, D., & Tsitsiklis, J. (2015). *Parallel and distributed computation: Numerical methods*. Athena Scientific.

Bevrani, H., & Hiyama, T. (2017). *Intelligent automatic generation control*. CRC Press.

Bhat, S. P., & Bernstein, D. S. (2000). Finite-time stability of continuous autonomous systems. *SIAM Journal on Control and Optimization*, *38*(3), 751–766.

Boqiang, R., & Chuanwen, J. (2009). A review on the economic dispatch and risk management considering wind power in the power market. *Renewable and Sustainable Energy Reviews*, *13*(8), 2169–2174.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine learning*, *3*(1), 1–122.

Bullo, F., Cortés, J., & Martinez, S. (2009). *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*: *vol. 27*, Princeton University Press.

Cao, W., Zhang, J., & Ren, W. (2015). Leader–follower consensus of linear multi-agent systems with unknown external disturbances. *Systems & Control Letters*, *82*, 64–70.

Chang, T. (2016). A proximal dual consensus ADMM method for multi-agent constrained optimization. *IEEE Transactions on Signal Processing*, *64*(14), 3719–3734.

Chang, T., Hong, M., & Wang, X. (2014). Multi-agent distributed optimization via inexact consensus ADMM. *IEEE Transactions on Signal Processing*, *63*(2), 482–497.

Chen, G., & Guo, Z. (2020). Initialization-free distributed fixed-time convergent algorithms for optimal resource allocation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *52*(2), 845–854.

Chen, G., & Li, Z. (2021). Distributed optimal resource allocation over strongly connected digraphs: A surplus-based approach. *Automatica*, *125*, Article 109459.

Chen, G., Ren, J., & Feng, E. N. (2016). Distributed finite-time economic dispatch of a network of energy resources. *IEEE Transactions on Smart Grid*, *8*(2), 822–832.

Chen, J., & Sayed, A. H. (2012). Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Transactions on Signal Processing*, *60*(8), 4289–4305.

Chen, J., Tu, S.-Y., & Sayed, A. H. (2011). Distributed optimization via diffusion adaptation. In *4th IEEE international workshop on computational advances in multi-sensor adaptive processing* (pp. 281–284). IEEE.

Chen, X., Yang, L., Chen, Z., Min, G., Zheng, X., & Rong, C. (2022). Resource allocation with workload-time windows for cloud-based software services: a deep reinforcement learning approach. *IEEE Transactions on Cloud Computing*.

Chen, G., & Zhao, Z. (2017). Delay effects on consensus-based distributed economic dispatch algorithm in microgrid. *IEEE Transactions on Power Systems*, *33*(1), 602–612.

Cherukuri, A., & Cortés, J. (2015). Distributed generator coordination for initialization and anytime optimization in economic dispatch. *IEEE Transactions on Control of Network Systems*, *2*(3), 226–237.

Cherukuri, A., Martínez, S., & Cortés, J. (2014). Distributed, anytime optimization in power-generator networks for economic dispatch. In *American control conference* (pp. 172–177). IEEE.

Chiang, M., Low, S. H., Calderbank, A. R., & Doyle, J. C. (2007). Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, *95*(1), 255–312.

Chiang, M., & Zhang, T. (2016). Fog and IoT: An overview of research opportunities. *IEEE Internet of Things Journal*, *3*(6), 854–864.

Chowdhury, B., & Rahman, S. (1990). A review of recent advances in economic dispatch. *IEEE Transactions on Power Systems*, *5*(4), 1248–1259.

Cortes, J. (2006). Finite-time convergent gradient flows with applications to network consensus. *Automatica*, *42*(11), 1993–2000.

Cortes, J. (2008). Discontinuous dynamical systems. *IEEE Control Systems Magazine*, *28*(3), 36–73.

Cortés, J., Martínez, S., & Bullo, F. (2006). Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, *51*(8), 1289–1298.

Cortés, J., Martınez, S., Karatas, T., & Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, *20*(2).

Dai, H., Fang, X., & Jia, J. (2022). Consensus-based distributed fixed-time optimization for a class of resource allocation problems. *Journal of the Franklin Institute*, *359*(18), 11135–11154.

Dallinger, D., & Wietschel, M. (2012). Grid integration of intermittent renewable energy sources using price-responsive plug-in electric vehicles. *Renewable and Sustainable Energy Reviews*, *16*(5), 3370–3382.

Dauphin, Y., De Vries, H., & Bengio, Y. (2015). Equilibrated adaptive learning rates for non-convex optimization. *Advances in Neural Information Processing Systems*, *28*.

Deghat, M., Ugrinovskii, V., Shames, I., & Langbort, C. (2019). Detection and mitigation of biasing attacks on distributed estimation networks. *Automatica*, *99*, 369–381.

Deng, W., Lai, M., Peng, Z., & Yin, W. (2017). Parallel multi-block ADMM with O(1/k) convergence. *Journal of Scientific Computing*, *71*, 712–736.

Deori, L., Margellos, K., & Prandini, M. (2016). On decentralized convex optimization in a multi-agent setting with separable constraints and its application to optimal charging of electric vehicles. In *IEEE 55th conference on decision and control* (pp. 6044–6049). IEEE.

Dibaji, S. M., Pirani, M., Flamholz, D. B., Annaswamy, A. M., Johansson, K. H., & Chakrabortty, A. (2019). A systems and control perspective of CPS security. *Annual Reviews in Control*, *47*, 394–411.

Doan, T. T., & Beck, C. L. (2017). Distributed Lagrangian methods for network resource allocation. In *IEEE conference on control technology and applications* (pp. 650–655).

Doan, T. T., & Olshevsky, A. (2017). Distributed resource allocation on dynamic networks in quadratic time. *Systems & Control Letters*, *99*, 57–63.

Doostmohammadian, M. (2020). Single-bit consensus with finite-time convergence: Theory and applications. *IEEE Transactions on Aerospace and Electronic Systems*, *56*(4), 3332–3338.

Doostmohammadian, M. (2023). Distributed energy resource management: All-time resource-demand feasibility, delay-tolerance, nonlinearity, and beyond. *IEEE Control Systems Letters*.

Doostmohammadian, M., & Aghasi, A. (2023). Robust-to-noise algorithms for distributed resource allocation and scheduling. In *11th RSI international conference on robotics and mechatronics* (pp. 15–20). IEEE.

Doostmohammadian, M., & Aghasi, A. (2024). Accelerated distributed allocation. *IEEE Signal Processing Letters*.

Doostmohammadian, M., Aghasi, A., Charalambous, T., & Khan, U. A. (2021). Distributed support vector machine over dynamic balanced directed networks. *IEEE Control Systems Letters*, *6*, 758–763.

Doostmohammadian, M., Aghasi, A., Pirani, M., Nekouei, E., Khan, U. A., & Charalambous, T. (2022). Fast-convergent anytime-feasible dynamics for distributed allocation of resources over switching sparse networks with quantized communication links. In *European control conference* (pp. 84–89).

Doostmohammadian, M., Aghasi, A., Rikos, A. I., Grammenos, A., Kalyvianaki, E., Hadjicostis, C. N., et al. (2022). Distributed anytime-feasible resource allocation subject to heterogeneous time-varying delays. *IEEE Open Journal of Control Systems*, *1*, 255–267.

Doostmohammadian, M., Aghasi, A., Vrakopoulou, M., & Charalambous, T. (2022). 1St-order dynamics on nonlinear agents for resource allocation over uniformly-connected networks. In *IEEE conference on control technology and applications* (pp. 1184–1189).

Doostmohammadian, M., Aghasi, A., Vrakopoulou, M., Rabiee, H. R., Khan, U. A., & Charalambous, T. (2023). Distributed delay-tolerant strategies for equality-constraint sum-preserving resource allocation. *Systems & Control Letters*.

Doostmohammadian, M., Aghasi, A., & Zarrabi, H. (2023). D-SVM over networked systems with non-ideal linking conditions. *Iran Journal of Computer Science*, 1–12.

Doostmohammadian, M., Jiang, W., & Charalambous, T. (2022). DTAC-ADMM: Delay-tolerant augmented consensus ADMM-based algorithm for distributed resource allocation. In *IEEE 61st conference on decision and control* (pp. 308–315). IEEE.

Doostmohammadian, M., Jiang, W., Liaquat, M., Aghasi, A., & Zarrabi, H. (2024). Discretized distributed optimization over dynamic digraphs. *IEEE Transactions on Automation Science and Engineering*.

Doostmohammadian, M., & Khan, U. (2013). On the genericity properties in distributed estimation: Topology design and sensor placement. *IEEE Journal of Selected Topics in Signal Processing*, *7*(2), 195–204.

Doostmohammadian, M., & Khan, U. (2014). Graph-theoretic distributed inference in social networks. *IEEE Journal of Selected Topics in Signal Processing*, *8*(4), 613–623.

Doostmohammadian, M., Khan, U. A., & Aghasi, A. (2022). Distributed constraint-coupled optimization over unreliable networks. In *10th RSI international conference on robotics and mechatronics* (pp. 371–376). IEEE.

Doostmohammadian, M., Khan, U., Pirani, M., & Charalambous, T. (2021). Consensus-based distributed estimation in the presence of heterogeneous, time-invariant delays. *IEEE Control Systems Letters*, *6*, 1598–1603.

Doostmohammadian, M., & Meskin, N. (2020). Sensor fault detection and isolation via networked estimation: Full-rank dynamical systems. *IEEE Transactions on Control of Network Systems*, *8*(2), 987–996.

Doostmohammadian, M., Sayyaadi, H., & Moarref, M. (2009). A novel consensus protocol using facility location algorithms. In *IEEE conference on control applications & intelligent control* (pp. 914–919).

Doostmohammadian, M., Taghieh, A., & Zarrabi, H. (2021). Distributed estimation approach for tracking a mobile target via formation of UAVs. *IEEE Transactions on Automation Science and Engineering*, *19*(4), 3765–3776.

Doostmohammadian, M., Vrakopoulou, M., Aghasi, A., & Charalambous, T. (2022). Distributed finite-sum constrained optimization subject to nonlinearity on the node dynamics. In *IEEE 95th vehicular technology conference:(VTC2022-spring)* (pp. 1–6). IEEE.

Doostmohammadian, M., Zarrabi, H., & Charalambous, T. (2022). Sensor fault detection and isolation via networked estimation: rank-deficient dynamical systems. *International Journal of Control*, 1–18.

Elsayed, W., & El-Saadany, E. (2014). A fully decentralized approach for solving the economic dispatch problem. *IEEE Transactions on Power Systems*, *30*(4), 2179–2189.

Enyioha, C., Jadbabaie, A., Preciado, V., & Pappas, G. (2015). Distributed resource allocation for control of spreading processes. In *2015 European control conference* (pp. 2216–2221). IEEE.

Ercetin, O., & Tassiulas, L. (2003). Market-based resource allocation for content delivery in the internet. *Institute of Electrical and Electronics Engineers. Transactions on Computers*, *52*(12), 1573–1585.

Ergu, D., Kou, G., Peng, Y., Shi, Y., & Shi, Y. (2013). The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment. *Journal of Supercomputing*, *64*, 835–848.

Esmaili, M., & Rajabi, M. (2014). Optimal charging of plug-in electric vehicles observing power grid constraints. *IET Gener. Transm. Distrib.*, *8*(4), 583–590.

Esteki, A., & Kia, S. S. (2022). Distributed optimal resource allocation with time-varying quadratic cost functions and resources over switching agents. In *European control conference* (pp. 441–446). IEEE.

Fagnani, F., & Zampieri, S. (2009). Average consensus with packet drop communication. *SIAM Journal on Control and Optimization*, *48*(1), 102–133.

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

Falsone, A., Margellos, K., & Prandini, M. (2018). A distributed iterative algorithm for multi-agent MILPs: Finite-time feasibility and performance characterization. *IEEE Control Systems Letters*, *2*(4), 563–568.

Falsone, A., Margellos, K., Zizzo, J., Prandini, M., & Garatti, S. (2022). On the sensitivity of linear resource sharing problems to the arrival of new agents. *IEEE Transactions on Automatic Control*.

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

Feng, Z., & Hu, G. (2017). Finite-time distributed optimization with quadratic objective functions under uncertain information. In *IEEE 56th annual conference on decision and control* (pp. 208–213). IEEE.

Firouzbahrami, M., & Nobakhti, A. (2022a). Cooperative fixed-time/finite-time distributed robust optimization of multi-agent systems. *Automatica*, *142*, Article 110358.

Firouzbahrami, M., & Nobakhti, A. (2022b). Finite-time distributed economic dispatch over network systems with coupled local costs. *IEEE Control Systems Letters*, *7*, 325–330.

Fu, W., Ma, Q., Qin, J., & Kang, Y. (2021). Resilient consensus-based distributed optimization under deception attacks. *International Journal of Robust and Nonlinear Control*, *31*(6), 1803–1816.

Gao, R., & Niu, H. (2021). A priority-based ADMM approach for flexible train scheduling problems. *Transportation Research Part C (Emerging Technologies)*, *123*, Article 102960.

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

Garg, K., & Panagou, D. (2021). Fixed-time stable gradient flows: Applications to continuous-time optimization. *IEEE Transactions on Automatic Control*, *66*(5), 2002–2015.

Ghadikolaei, H. S., & Magnússon, S. (2020). Communication-efficient variance-reduced stochastic gradient descent. *IFAC-PapersOnLine*, *53*(2), 2648–2653.

Ghadimi, E., Johansson, M., & Shames, I. (2011). Accelerated gradient methods for networked optimization. In *IEEE American control conference* (pp. 1668–1673).

Gharesifard, B., & Cortés, J. (2012). Distributed strategies for generating weight-balanced and doubly stochastic digraphs. *European Journal of Control*, *18*(6), 539–557.

Gharesifard, B., & Cortés, J. (2013). Distributed continuous-time convex optimization on weight-balanced digraphs. *IEEE Transactions on Automatic Control*, *59*(3), 781–786.
Giraldo, J., Urbina, D., Cardenas, A., Valente, J., Faisal, M., Ruths, J., et al. (2018). A survey of physics-based attack detection in cyber-physical systems. *ACM Computing Surveys*, *51*(4), 1–36.
Godsil, C., & Royle, G. (2001). *Algebraic graph theory*. New York: Springer.
Gong, X., Cui, Y., Shen, J., Xiong, J., & Huang, T. (2022). Distributed optimization in prescribed-time: Theory and experiment. *IEEE Transactions on Network Science and Engineering*, *9*(2), 564–576.
Grammenos, A., Charalambous, T., & Kalyvianaki, E. (2021). CPU scheduling in data centers using asynchronous finite-time distributed coordination mechanisms. arXiv preprint arXiv:2101.06139.
Grammenos, A., Charalambous, T., & Kalyvianaki, E. (2023). CPU scheduling in data centers using asynchronous finite-time distributed coordination mechanisms. *IEEE Transactions on Network Science and Engineering*.
Gross, J. L., & Yellen, J. (2004). *Handbook of graph theory*. CRC Press.
Gu, Y., She, C., Quan, Z., Qiu, C., & Xu, X. (2023). Graph neural networks for distributed power allocation in wireless networks: Aggregation over-the-air. *IEEE Transactions on Wireless Communication*, *22*(11), 7551–7564.
Gu, B., Zhang, X., Lin, Z., & Alazab, M. (2020). Deep multiagent reinforcement-learning-based resource allocation for internet of controllable things. *IEEE Internet of Things Journal*, *8*(5), 3066–3074.
Gündoğan, A., Gürsu, H. M., Pauli, V., & Kellerer, W. (2020). Distributed resource allocation with multi-agent deep reinforcement learning for 5G-V2V communication. In *21st international symposium on theory, algorithmic foundations, and protocol design for mobile networks and mobile computing* (pp. 357–362).
Guo, Z., & Chen, G. (2022). Distributed dynamic event-triggered and practical predefined-time resource allocation in cyber–physical systems. *Automatica*, *142*, Article 110390.
Guo, S., Liu, J., Yang, Y., Xiao, B., & Li, Z. (2018). Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing. *IEEE Transactions on Mobile Computing*, *18*(2), 319–333.
Guo, Y., Zhao, R., Lai, S., Fan, L., Lei, X., & Karagiannidis, G. K. (2022). Distributed machine learning for multiuser mobile edge computing systems. *IEEE Journal of Selected Topics in Signal Processing*, *16*(3), 460–473.
Haddadpour, F., Kamani, M., Mahdavi, M., & Cadambe, V. (2019). Trading redundancy for communication: Speeding up distributed SGD for non-convex optimization. In *International conference on machine learning* (pp. 2545–2554). PMLR.
Hadjicostis, C. N., & Charalambous, T. (2011). Asynchronous coordination of distributed energy resources for the provisioning of ancillary services. In *49th IEEE allerton conference on communication, control, and computing* (pp. 1500–1507).
Hadjicostis, C. N., & Charalambous, T. (2014). Average consensus in the presence of delays in directed graph topologies. *IEEE Transactions on Automatic Control*, *59*(3), 763–768.
Hadjicostis, C. N., Domínguez-García, A., Charalambous, T., et al. (2018). Distributed averaging and balancing in network systems: with applications to coordination and control. *Foundations and Trends® in Systems and Control*, *5*(2–3), 99–292.
Hadjicostis, C., Vaidya, N., & Domínguez-García, A. (2015). Robust distributed average consensus via exchange of running sums. *IEEE Transactions on Automatic Control*, *61*(6), 1492–1507.
Hadjicostis, C. N., Vaidya, N. H., & Domínguez-García, A. D. (2016). Robust distributed average consensus via exchange of running sums. *IEEE Transactions on Automatic Control*, *61*(6), 1492–1507.
Hasan, N., Ejaz, W., Ejaz, N., Kim, H. S., Anpalagan, A., & Jo, M. (2016). Network selection and channel allocation for spectrum sharing in 5G heterogeneous networks. *IEEE Access*, *4*, 980–992.
Hattab, G., Ucar, S., Higuchi, T., Altintas, O., Dressler, F., & Cabric, D. (2019). Optimized assignment of computational tasks in vehicular micro clouds. In *2nd international workshop on edge systems, analytics and networking* (pp. 1–6).
Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd annual hawaii international conference on system sciences* (p. 10). IEEE.
Hu, Z., & Yang, J. (2018). Distributed finite-time optimization for second order continuous-time multiple agents systems with time-varying cost function. *Neurocomputing*, *287*, 173–184.
Huang, Y., Meng, Z., Sun, J., & Ren, W. (2023). Distributed multi-proximal algorithm for nonsmooth convex optimization with coupled inequality constraints. *IEEE Transactions on Automatic Control*, 1–8.
Huang, Y., Meng, Z., Sun, J., & Wang, G. (2024). Distributed continuous-time proximal algorithm for nonsmooth resource allocation problem with coupled constraints. *Automatica*, *159*, Article 111309.
Hussain, F., Hussain, R., Anpalagan, A., & Benslimane, A. (2020). A new block-based reinforcement learning approach for distributed resource allocation in clustered IoT networks. *IEEE Transactions on Vehicular Technology*, *69*(3), 2891–2904.
Ioan, D., Prodan, I., Olaru, S., Stoican, F., & Niculescu, S. (2021). Mixed-integer programming in motion planning. *Annual Reviews in Control*, *51*, 65–87.
Iouditski, A. (2015). *Lecture notes*, *Convex optimization I: Introduction*. Laboratoire Jean Kuntzmann, FR.
Iutzeler, F., Bianchi, P., Ciblat, P., & Hachem, W. (2015). Explicit convergence rate of a distributed alternating direction method of multipliers. *IEEE Transactions on Automatic Control*, *61*(4), 892–904.
Jacquot, P., Beaude, O., Benchimol, P., Gaubert, S., & Oudjane, N. (2019). A privacy-preserving disaggregation algorithm for non-intrusive management of flexible energy. In *IEEE 58th conference on decision and control* (pp. 890–896). IEEE.
Jadbabaie, A., Lin, J., & Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, *48*(6), 988–1001.
Jayakumar, S., & Nandakumar, S. (2023). Reinforcement learning based distributed resource allocation technique in device-to-device (D2D) communication. *Wireless Networks*, *29*(4), 1843–1858.
Ji, Z., & Qin, Z. (2022). Federated learning for distributed energy-efficient resource allocation. In *IEEE international conference on communications* (pp. 1–6). IEEE.
Ji, Z., Qin, Z., & Tao, X. (2023). Meta federated reinforcement learning for distributed resource allocation. *IEEE Transactions on Wireless Communication*.
Ji, L., Yu, L., Zhang, C., Guo, X., & Li, H. (2023). Initialization-free distributed prescribed-time consensus based algorithm for economic dispatch problem over directed network. *Neurocomputing*, *533*, 1–9.
Jian, L., Hu, J., Wang, J., & Shi, K. (2019). Distributed inexact dual consensus ADMM for network resource allocation. *Optimal Control Applications & Methods*, *40*(6), 1071–1087.
Jiang, W., & Charalambous, T. (2021). Distributed alternating direction method of multipliers using finite-time exact ratio consensus in digraphs. In *European control conference* (pp. 2205–2212). IEEE.
Jiang, W., Doostmohammadian, M., & Charalambous, T. (2022). Distributed resource allocation via ADMM over digraphs. In *IEEE 61st conference on decision and control* (pp. 5645–5651). IEEE.
Kalyvianaki, E., Charalambous, T., & Hand, S. (2009). Self-adaptive and self-configured CPU resource provisioning for virtualized servers using Kalman filters. In *Proceedings of the 6th international conference on autonomic computing* (pp. 117–126).
Kar, S., & Hug, G. (2012). Distributed robust economic dispatch in power systems: A consensus + innovations approach. In *IEEE power and energy society general meeting* (pp. 1–8).
Kar, S., & Moura, J. M. F. (2013). Consensus+ innovations distributed inference over networks: cooperation and sensing in networked systems. *IEEE Signal Processing Magazine*, *30*(3), 99–109.
Kelly, F. P., Maulloo, A. K., & Tan, D. K. H. (1998). Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research society*, *49*, 237–252.
Khan, U. A., Bajwa, W. U., Nedić, A., Rabbat, M. G., & Sayed, A. H. (2020). Optimization for data-driven learning and control. *Proceedings of the IEEE*, *108*(11), 1863–1868.
Kia, S. S. (2017). Distributed optimal in-network resource allocation algorithm design via a control theoretic approach. *Systems & Control Letters*, *107*, 49–57.
Kia, S. S., Cortés, J., & Martínez, S. (2015). Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication. *Automatica*, *55*, 254–264.
Kia, S. S., Van Scoy, B., Cortes, J., Freeman, R. A., Lynch, K. M., & Martinez, S. (2019). Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems Magazine*, *39*(3), 40–72.
Kisacikoglu, M., Erden, F., & Erdogan, N. (2017). Distributed control of PEV charging based on energy demand forecast. *IEEE Transactions on Industrial Informatics*, *14*(1), 332–341.
Köhler, J., Müller, M. A., Li, N., & Allgöwer, F. (2017). Real time economic dispatch for power networks: A distributed economic model predictive control approach. In *IEEE 56th annual conference on decision and control* (pp. 6340–6345). IEEE.
Kraning, M., Chu, E., Lavaei, J., & Boyd, S. (2013). Dynamic network energy management via proximal message passing. *Foundations and Trends® in Optimization*, *1*(2), 73–126.
Kumar, H., Chauhan, N. K., & Yadav, P. K. (2021). A high performance model for task allocation in distributed computing system using k-means clustering technique. In *Research anthology on architectures, frameworks, and integration strategies for distributed and cloud computing* (pp. 1244–1268). IGI Global.
Kumar, Y., Kaul, S., & Hu, Y. (2022). Machine learning for energy-resource allocation, workflow scheduling and live migration in cloud computing: State-of-the-art survey. *Sustainable Computing: Informatics and Systems*, *36*, Article 100780.
Kumar, P., & Kothari, D. (2005). Recent philosophies of automatic generation control strategies in power systems. *IEEE Transactions on Power Systems*, *20*(1), 346–357.
Kuwaranancharoen, K., Xin, L., & Sundaram, S. (2020). Byzantine-resilient distributed optimization of multi-dimensional functions. In *American control conference* (pp. 4399–4404). IEEE.
Lakshmanan, H., & De Farias, D. P. (2008). Decentralized resource allocation in dynamic networks of agents. *SIAM Journal on Optimization*, *19*(2), 911–940.
Lesser, V., Ortiz, C. L., Jr., & Tambe, M. (2003). *Distributed sensor networks: A multiagent perspective*: *vol. 9*, Springer Science & Business Media.
Li, B., Cen, S., Chen, Y., & Chi, Y. (2020). Communication-efficient distributed optimization in networks with gradient tracking and variance reduction. *Journal of Machine Learning Research*, *21*(1), 7331–7381.
Li, Z., & Ding, Z. (2020). Distributed multiobjective optimization for network resource allocation of multiagent systems. *IEEE Transactions on Cybernetics*, *51*(12), 5800–5810.

Li, M., Gao, J., Chen, N., Zhao, L., & Shen, X. (2019). Decentralized PEV power allocation with power distribution and transportation constraints. *IEEE Journal on Selected Areas in Communications*, *38*(1), 229–243.

Li, L., Ho, D., & Lu, J. (2016). A consensus recovery approach to nonlinear multi-agent system under node failure. *Information Sciences*, *367*, 975–989.

Li, W., Lin, Z., Cai, K., & Yan, G. (2020). Distributed algorithm for a finite time horizon resource allocation over a directed network. *IET Control Theory & Applications*, *14*(9), 1170–1182.

Li, J., & Su, H. (2022). Implicit tracking-based distributed constraint-coupled optimization. *IEEE Transactions on Control of Network Systems*.

Li, X., Yi, X., & Xie, L. (2020). Distributed online optimization for multi-agent networks with coupled inequality constraints. *IEEE Transactions on Automatic Control*, *66*(8), 3575–3591.

Li, N., Zhao, C., & Chen, L. (2015). Connecting automatic generation control and economic dispatch from an optimization view. *IEEE Transactions on Control of Network Systems*, *3*(3), 254–264.

Lim, W. B., Ng, J., Xiong, Z., Niyato, D., Miao, C., & Kim, D. (2021). Dynamic edge association and resource allocation in self-organizing hierarchical federated learning networks. *IEEE Journal on Selected Areas in Communications*, *39*(12), 3640–3653.

Lin, P., Ren, W., & Farrell, J. A. (2017). Distributed continuous-time optimization: nonuniform gradient gains, finite-time convergence, and convex constraint set. *IEEE Transactions on Automatic Control*, *62*(5), 2239–2253.

Lin, X., Shroff, N. B., & Srikant, R. (2006). A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected areas in Communications*, *24*(8), 1452–1463.

Lin, W., Wang, Y., Li, C., & Yu, X. (2020a). Distributed resource allocation: an indirect dual ascent method with an exponential convergence rate. *Nonlinear Dynamics*, *102*, 1685–1699.

Lin, W., Wang, Y., Li, C., & Yu, X. (2020b). Predefined-time optimization for distributed resource allocation. *Journal of the Franklin Institute*, *357*(16), 11323–11348.

Lin, W., Wang, Y., Li, C., & Yu, X. (2021). Distributed resource allocation via accelerated saddle point dynamics. *IEEE/CAA Journal of Automatica Sinica*, *8*(9), 1588–1599.

Ling, Q., & Ribeiro, A. (2013). Decentralized dynamic optimization through the alternating direction method of multipliers. *IEEE Transactions on Signal Processing*, *62*(5), 1185–1197.

Liu, X., Jia, M., & Ding, H. (2020). Uplink resource allocation for multicarrier grouping cognitive internet of things based on K-means learning. *Ad Hoc Networks*, *96*, Article 102002.

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., et al. (2019). On the variance of the adaptive learning rate and beyond. In *International conference on learning representations*.

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

Liu, Z., You, X., Yang, H., & Zhao, L. (2015). Leader-following consensus of heterogeneous multi-agent systems with packet dropout. *International Journal of Control, Automation and Systems*, *13*, 1067–1075.

Makhdoumi, A., & Ozdaglar, A. (2017). Convergence rate of distributed ADMM over networks. *IEEE Transactions on Automatic Control*, *62*(10), 5082–5095.

Makridis, E., Deliparaschos, K., Kalyvianaki, E., Zolotas, A., & Charalambous, T. (2020). Robust dynamic CPU resource provisioning in virtualized servers. *IEEE Transactions on Services Computing*, *15*(2), 956–969.

Malairajan, R. A., Ganesh, K., Muhos, M., & Anbuudayasankar, S. P. (2013). Class of resource allocation problems in supply chain–a review. *International Journal of Business Innovation and Research*, *7*(1), 113–139.

Mann, Z. (2015). Allocation of virtual machines in cloud data centers–a survey of problem models and optimization algorithms. *ACM Computing Surveys (CSUR)*, *48*(1), 1–34.

Mirtabatabaei, A., & Bullo, F. (2012). Opinion dynamics in heterogeneous networks: Convergence conjectures and theorems. *SIAM Journal on Control and Optimization*, *50*(5), 2763–2785.

Mo, Y., & Murray, R. M. (2016). Privacy preserving average consensus. *IEEE Transactions on Automatic Control*, *62*(2), 753–765.

Mohajer, A., Barari, M., & Zarrabi, H. (2017). Big data based self-optimization networking: A novel approach beyond cognition. *Intelligent Automation & Soft Computing*, 1–7.

Molzahn, D. K., Dörfler, F., Sandberg, H., Low, S. H., Chakrabarti, S., Baldick, R., et al. (2017). A survey of distributed optimization and control algorithms for electric power systems. *IEEE Transactions on Smart Grid*, *8*(6), 2941–2962.

Morariu, C., Morariu, O., Răileanu, S., & Borangiu, T. (2020). Machine learning for predictive scheduling and resource allocation in large scale manufacturing systems. *Computers in Industry*, *120*, Article 103244.

Mukherjee, J., & Gupta, A. (2016). Distributed charge scheduling of plug-in electric vehicles using inter-aggregator collaboration. *IEEE Transactions on Smart Grid*, *8*(1), 331–341.

Nasir, Y., & Guo, D. (2019). Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks. *IEEE Journal on Selected Areas in Communications*, *37*(10), 2239–2250.

Nedic, A. (2020). Distributed gradient methods for convex machine learning problems in networks: Distributed optimization. *IEEE Signal Processing Magazine*, *37*(3), 92–101.

Nedić, A., & Liu, J. (2018). Distributed optimization for control. *Annual Review of Control, Robotics, and Autonomous Systems*, *1*, 77–103.

Nedić, A., & Olshevsky, A. (2014). Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, *60*(3), 601–615.

Nedic, A., Olshevsky, A., Ozdaglar, A., & Tsitsiklis, J. N. (2008). Distributed subgradient methods and quantization effects. In *47th IEEE conference on decision and control* (pp. 4177–4184).

Nedic, A., Olshevsky, A., & Shi, W. (2017). Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, *27*(4), 2597–2633.

Nedić, A., Olshevsky, A., & Shi, W. (2018). Improved convergence rates for distributed resource allocation. In *IEEE conference on decision and control* (pp. 172–177). IEEE.

Nedić, A., Olshevsky, A., Shi, W., & Uribe, C. (2017). Geometrically convergent distributed optimization with uncoordinated step-sizes. In *American control conference* (pp. 3950–3955). IEEE.

Nedic, A., & Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, *54*(1), 48–61.

Nekouei, E., Alpcan, T., Nair, G. N., & Evans, R. J. (2016). Convergence analysis of quantized primal-dual algorithms in network utility maximization problems. *IEEE Transactions on Control of Network Systems*, *5*(1), 284–297.

Nekouei, E., Nair, G. N., & Alpcan, T. (2016). Performance analysis of gradient-based nash seeking algorithms under quantization. *IEEE Transactions on Automatic Control*, *61*(12), 3771–3783.

Nekouei, E., Tanaka, T., Skoglund, M., & Johansson, K. H. (2019). Information-theoretic approaches to privacy in estimation and control. *Annual Reviews in Control*, *47*, 412–422.

Nesterov, Y. (1998). Introductory lectures on convex programming, volume I: Basic course. *Lecture Notes*, *3*(4), 5.

Nguyen, V., Sharma, S., Vu, T., Chatzinotas, S., & Ottersten, B. (2020). Efficient federated learning algorithm for resource allocation in wireless IoT networks. *IEEE Internet of Things Journal*, *8*(5), 3394–3409.

Nie, Y., Zhao, J., Gao, F., & Yu, F. R. (2021). Semi-distributed resource management in UAV-aided MEC systems: A multi-agent federated reinforcement learning approach. *IEEE Transactions on Vehicular Technology*, *70*(12), 13162–13173.

Ning, B., Han, Q., & Zuo, Z. (2017). Distributed optimization for multiagent systems: An edge-based fixed-time consensus approach. *IEEE Transactions on Cybernetics*, *49*(1), 122–132.

Nowzari, C., Preciado, V. M., & Pappas, G. J. (2016). Analysis and control of epidemics: A survey of spreading processes on complex networks. *IEEE Control Systems Magazine*, *36*(1), 26–46.

Ochoa, D. E., Poveda, J. I., Uribe, C. A., & Quijano, N. (2019). Hybrid robust optimal resource allocation with momentum. In *58th conference on decision and control* (pp. 3954–3959).

Olfati-Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, *95*(1), 215–233.

Olfati-Saber, R., & Jalalkamali, P. (2011). Collaborative target tracking using distributed Kalman filtering on mobile sensor networks. In *American control conference*. San Francisco, CA.

Olfati-Saber, R., & Murray, R. M. (2002). Distributed cooperative control of multiple vehicle formations using structural potential functions. *IFAC Proceedings Volumes*, *35*(1), 495–500.

Olfati-Saber, R., & Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, *49*(9), 1520–1533.

Olshevsky, A. (2015). Linear time average consensus on fixed graphs. *IFAC-PapersOnLine*, *48*(22), 94–99.

Palomar, D. P., & Chiang, M. (2006). A tutorial on decomposition methods for network utility maximization. *IEEE Journal on Selected Areas in Communications*, *24*(8), 1439–1451.

Park, S., & Martins, N. (2012). Necessary and sufficient conditions for the stabilizability of a class of LTI distributed observers. In *51st IEEE conference on decision and control* (pp. 7431–7436).

Parsegov, S. E., Polyakov, A. E., & Shcherbakov, P. S. (2013). Fixed-time consensus algorithm for multi-agent systems with integrator dynamics. *IFAC Proceedings Volumes*, *46*(27), 110–115.

Pasqualetti, F., Dorfler, F., & Bullo, F. (2013). Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, *58*(11), 2715–2729.

Pelikan, M., Sastry, K., & Cantú-Paz, E. (2006). *Scalable optimization via probabilistic modeling: From algorithms to applications*. Springer.

Pequito, S., Kar, S., Sundaram, S., & Aguiar, A. P. (2014). Design of communication networks for distributed computation with privacy guarantees. In *IEEE 53rd annual conference on decision and control* (pp. 1370–1376).

Peteiro-Barral, D., & Guijarro-Berdiñas, B. (2013). A survey of methods for distributed machine learning. *Progress in Artificial Intelligence*, *2*, 1–11.

Pirani, M., Hashemi, E., Khajepour, A., Fidan, B., Litkouhi, B., Chen, S., et al. (2018). Cooperative vehicle speed fault diagnosis and correction. *IEEE Transactions on Intelligent Transportation Systems*, *20*(2), 783–789.

Pourbabak, H., Luo, J., Chen, T., & Su, W. (2017). A novel consensus-based distributed algorithm for economic dispatch based on local estimation of power mismatch. *IEEE Transactions on Smart Grid*, *9*(6), 5930–5942.

Qureshi, M. I., Xin, R., Kar, S., & Khan, U. A. (2020). S-ADDOPT: Decentralized stochastic first-order optimization over directed graphs. *IEEE Control Systems Letters*, *5*(3), 953–958.

Qureshi, M., Xin, R., Kar, S., & Khan, U. A. (2021). Push-SAGA: A decentralized stochastic algorithm with variance reduction over directed graphs. *IEEE Control Systems Letters*, *6*, 1202–1207.

Ramseyer, G., Goel, A., & Mazières, D. (2023). SPEEDEX: A scalable, parallelizable, and economically efficient decentralized EXchange. In *20th USENIX symposium on networked systems design and implementation* (pp. 849–875).

Ren, W., & Beard, R. W. (2005). Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, *50*(5), 655–661.

Ren, W., Beard, R. W., & Atkins, E. M. (2005). A survey of consensus problems in multi-agent coordination. In *American control conference* (pp. 1859–1864). Portland, OR.

Rikos, A. I., Charalambous, T., Johansson, K. H., & Hadjicostis, C. N. (2020). Privacy-preserving event-triggered quantized average consensus. In *59th IEEE conference on decision and control* (pp. 6246–6253).

Rikos, A. I., Grammenos, A., Kalyvianaki, E., Hadjicostis, C. N., Charalambous, T., & Johansson, K. H. (2021). Optimal CPU scheduling in data centers via a finite-time distributed quantized coordination mechanism. In *60th IEEE conference on decision and control* (pp. 6276–6281).

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

Rikos, A. I., Nylöf, J., Gracy, S., & Johansson, K. H. (2022). Distributed optimal allocation with quantized communication and privacy-preserving guarantees. *IFAC-PapersOnLine*, *55*(41), 64–70.

Ruan, M., Gao, H., & Wang, Y. (2019). Secure and privacy-preserving consensus. *IEEE Transactions on Automatic Control*, *64*(10), 4035–4049.

Sardana, A., & Joshi, R. (2009). An auto-responsive honeypot architecture for dynamic resource allocation and QoS adaptation in ddos attacked networks. *Computer Communications*, *32*(12), 1384–1399.

Sayyaadi, H., & Doostmohammadian, M. (2011). Finite-time consensus in directed switching network topologies and time-delayed communications. *Scientia Iranica*, *18*(1), 75–85.

Sayyaadi, H., & Moarref, M. (2011). A distributed algorithm for proportional task allocation in networks of mobile agents. *IEEE Transactions on Automatic Control*, *56*(2), 405–410.

Scaman, K., Bach, F., Bubeck, S., Lee, Y., & Massoulié, L. (2017). Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *International conference on machine learning* (pp. 3027–3036). PMLR.

Schenato, L., & Fiorentin, F. (2011). Average timesynch: A consensus-based protocol for clock synchronization in wireless sensor networks. *Automatica*, *47*(9), 1878–1886.

Schenato, L., & Gamba, G. (2007). A distributed consensus protocol for clock synchronization in wireless sensor network. In *46th IEEE conference on decision and control* (pp. 2289–2294). IEEE.

Seuret, A., Dimarogonas, D., & Johansson, K. H. (2008). Consensus under communication delays. In *47th IEEE conference on decision and control* (pp. 4922–4927). IEEE.

Shang, Y. (2017). Fixed-time group consensus for multi-agent systems with non-linear dynamics and uncertainties. *IET Control Theory & Applications*.

Shao, G., Wang, R., Wang, X., & Liu, K. (2020). Distributed algorithm for resource allocation problems under persistent attacks. *Journal of the Franklin Institute*, *357*(10), 6241–6256.

Shi, X., Su, Y., Mu, C., & Sun, C. (2024). Nonsmooth distributed resource allocation over second-order nonlinear multiagent systems. *IEEE Systems Journal*.

Shi, X., Xu, L., Yang, T., Lin, Z., & Wang, X. (2022). Distributed fixed-time resource allocation algorithm for the general linear multi-agent systems. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *69*(6), 2867–2871.

Skobelev, P. (2015). Multi-agent systems for real-time adaptive resource management. In *Industrial agents* (pp. 207–229). Elsevier.

Song, Y., Cao, J., & Rutkowski, L. (2021). A fixed-time distributed optimization algorithm based on event-triggered strategy. *IEEE Transactions on Network Science and Engineering*, *9*(3), 1154–1162.

Srikant, R., & Ying, L. (2013). *Communication networks: an optimization, control, and stochastic networks perspective*. Cambridge University Press.

Stanković, S. S., Beko, M., & Stanković, M. S. (2019). Robust nonlinear consensus seeking. In *58th IEEE conference on decision and control* (pp. 4465–4470).

Stich, S. (2019). Local SGD converges fast and communicates little. In *2019-international conference on learning representations*.

Sun, S., Yang, Q., & Yan, W. (2017). Optimal temporal-spatial PEV charging scheduling in active power distribution networks. *Protection and Control of Modern Power Systems*, *2*, 1–10.

Sundaram, S., & Gharesifard, B. (2016). Secure local filtering algorithms for distributed optimization. In *IEEE 55th conference on decision and control* (pp. 1871–1876). IEEE.

Sundaram, S., & Hadjicostis, C. (2008). Distributed function calculation and consensus using linear iterative strategies. *IEEE Journal on Selected Areas in Communications*, [ISSN: 0733-8716] *26*(4), 650–660.

Sundhar Ram, S., Nedić, A., & Veeravalli, V. V. (2012). A new class of distributed optimization algorithms: Application to regression of distributed data. *Optimization Methods & Software*, *27*(1), 71–88.

Taillard, E. D., Laporte, G., & Gendreau, M. (1996). Vehicle routeing with multiple use of vehicles. *Journal of the Operational Research Society*, *47*(8), 1065–1070.

Tanner, H., Jadbabaie, A., & Pappas, G. J. (2007). Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control*, *52*(5), 863–868.

Tanner, H. G., Pappas, G. J., & Kumar, V. (2004). Leader-to-formation stability. *IEEE Transactions on Robotics and Automation*, *20*(3), 433–455.

Telsang, B. (2022). *Decentralized resource allocation through constrained centroidal voronoi tessellations* (Ph.D. thesis), The University of Tennessee – Knoxville.

Trossen, D., & Parisis, G. (2012). Designing and realizing an information-centric internet. *IEEE Communications Magazine*, *50*(7), 60–67.

Tsiropoulos, G. I., Dobre, O. A., Ahmed, M. H., & Baddour, K. E. (2014). Radio resource allocation techniques for efficient spectrum access in cognitive radio networks. *IEEE Communications Surveys & Tutorials*, *18*(1), 824–847.

Turan, B., Uribe, C. A., Wai, H., & Alizadeh, M. (2020). Resilient primal–dual optimization algorithms for distributed resource allocation. *IEEE Transactions on Control of Network Systems*, *8*(1), 282–294.

Turan, B., Uribe, C. A., Wai, H., & Alizadeh, M. (2021). Resilient prima-dual optimization algorithms for distributed resource allocation. *IEEE Transactions on Control of Network Systems*, *8*(1), 282–294.

Ullah, K., Basit, A., Ullah, Z., Aslam, S., & Herodotou, H. (2021). Automatic generation control strategies in conventional and modern power systems: A comprehensive overview. *Energies*, *14*(9), 2376.

Uribe, C. A., Lee, S., Gasnikov, A., & Nedic, A. (2020). A dual approach for optimal algorithms in distributed optimization over networks. In *Information theory and applications workshop* (pp. 1–37).

Vaidya, N. H., Hadjicostis, C. N., & Dominguez-Garcia, A. D. (2012a). Robust average consensus over packet dropping links: Analysis via coefficients of ergodicity. In *51st IEEE conference on decision and control* (pp. 2761–2766).

Vaidya, N., Hadjicostis, C., & Domínguez-García, A. (2012b). Robust average consensus over packet dropping links: Analysis via coefficients of ergodicity. In *IEEE 51st IEEE conference on decision and control* (pp. 2761–2766). IEEE.

Vinyals, M., Rodriguez-Aguilar, J., & Cerquides, J. (2011). A survey on sensor networks from a multiagent perspective. *The Computer Journal*, *54*(3), 455–470.

Vlahakis, E., Athanasopoulos, N., & McLoone, S. (2021). AIMD scheduling and resource allocation in distributed computing systems. In *60th IEEE conference on decision and control* (pp. 4642–4647). IEEE.

Vujanic, R., Esfahani, P. M., Goulart, P. J., Mariéthoz, S., & Morari, M. (2016). A decomposition method for large scale MILPs, with performance guarantees and a power system application. *Automatica*, *67*, 144–156.

Wang, Z., Eisen, M., & Ribeiro, A. (2021). Unsupervised learning for asynchronous resource allocation in ad-hoc wireless networks. In *IEEE international conference on acoustics, speech and signal processing* (pp. 8143–8147). IEEE.

Wang, Z., Eisen, M., & Ribeiro, A. (2022). Learning decentralized wireless resource allocations with graph neural networks. *IEEE Transactions on Signal Processing*, *70*, 1850–1863.

Wang, B., Fei, Q., & Wu, Q. (2020). Distributed time-varying resource allocation optimization based on finite-time consensus approach. *IEEE Control Systems Letters*, *5*(2), 599–604.

Wang, X., Hong, Y., Sun, X., & Liu, K. (2019). Distributed optimization for resource allocation problems under large delays. *IEEE Transactions on Industrial Electronics*, *66*(12), 9448–9457.

Wang, R., Liu, Y., & Ling, Q. (2022). Byzantine-resilient resource allocation over decentralized networks. *IEEE Transactions on Signal Processing*, *70*, 4711–4726.

Wang, Y., Song, Y., Hill, D. J., & Krstic, M. (2018). Prescribed-time consensus and containment control of networked multiagent systems. *IEEE Transactions on Cybernetics*, *49*(4), 1138–1147.

Wang, H., & Van Mieghem, P. (2008). Algebraic connectivity optimization via link addition. In *3rd international conference on bio-inspired models of network, information and computing systems* (pp. 1–8).

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

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

Wei, W., Gu, H., Wang, K., Li, J., Zhang, X., & Wang, N. (2022). Multi-dimensional resource allocation in distributed data centers using deep reinforcement learning. *IEEE Transactions on Network and Service Management*, *20*(2), 1817–1829.

Wei, E., & Ozdaglar, A. (2012). Distributed alternating direction method of multipliers. In *IEEE 51st IEEE conference on decision and control* (pp. 5445–5450). IEEE.

Wei, J., Yi, X., Sandberg, H., & Johansson, K. H. (2019). Nonlinear consensus protocols with applications to quantized communication and actuation. *IEEE Transactions on Control of Network Systems*, *6*(2), 598–608.

Wen, G., Yu, X., & Liu, Z. (2021). Recent progress on the study of distributed economic dispatch in smart grid: an overview. *Frontiers of Information Technology & Electronic Engineering*, *22*(1), 25–39.

Wood, A. J., Wollenberg, B. F., & Sheblé, G. B. (2013). *Power generation, operation, and control*. John Wiley & Sons.

Wu, W., Liu, S., & Zhu, S. (2022). Distributed dual gradient tracking for economic dispatch in power systems with noisy information. *Electric Power Systems Research*, *211*, Article 108298.

Wu, X., Magnusson, S., & Johansson, M. (2021). A new family of feasible methods for distributed resource allocation. In *60th IEEE conference on decision and control* (pp. 3355–3360).

Xia, X., & Elaiw, A. M. (2010). Optimal dynamic economic dispatch of generation: A review. *Electric Power Systems Research*, *80*(8), 975–986.

Xiao, L., & Boyd, S. (2006). Optimal scaling of a gradient method for distributed resource allocation. *Journal of Optimization Theory and Applications*, *129*(3), 469–488.

Xiao, F., & Wang, L. (2008). Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays. *IEEE Transactions on Automatic Control*, *53*(8), 1804–1816.

Xin, R., Kar, S., & Khan, U. A. (2020). Decentralized stochastic optimization and machine learning: A unified variance-reduction framework for robust performance and fast convergence. *IEEE Signal Processing Magazine*, *37*(3), 102–113.

Xin, R., & Khan, U. A. (2018). A linear algorithm for optimization over directed graphs with geometric convergence. *IEEE Control Systems Letters*, *2*(3), 315–320.

Xin, R., Sahu, A. K., Khan, U. A., & Kar, S. (2019). Distributed stochastic optimization with gradient tracking over strongly-connected networks. In *IEEE conference on decision and control* (pp. 8353–8358).

Xin, R., Xi, C., & Khan, U. A. (2019). FROST–fast row-stochastic optimization with uncoordinated step-sizes. *EURASIP Journal on Advances in Signal Processing*, *2019*(1), 1–14.

Xu, Y., Han, T., Cai, K., Lin, Z., Yan, G., & Fu, M. (2017). A distributed algorithm for resource allocation over dynamic digraphs. *IEEE Transactions on Signal Processing*, *65*(10), 2600–2612.

Xu, C., & Liu, Q. (2022). A resilient distributed optimization algorithm based on consensus of multi-agent system against two attack scenarios. *Journal of the Franklin Institute*.

Xu, C., Liu, Q., & Huang, T. (2022). Resilient penalty function method for distributed constrained optimization under byzantine attack. *Information Sciences*, *596*, 362–379.

Xu, J., Zhu, S., Soh, Y., & Xie, L. (2018). A dual splitting approach for distributed resource allocation with regularization. *IEEE Transactions on Control of Network Systems*, *6*(1), 403–414.

Yang, Y., Jia, Q., Xu, Z., Guan, X., & Spanos, C. J. (2022). Proximal ADMM for nonconvex and nonsmooth optimization. *Automatica*, *146*, Article 110551.

Yang, S., Tan, S., & Xu, J. (2013). Consensus based approach for economic dispatch problem in a smart grid. *IEEE Transactions on Power Systems*, *28*(4), 4416–4426.

Yang, P., Xu, M., Li, D., Liu, Z., & Huang, Y. (2019). Distributed fault tolerant consensus control for multi-agent system with actuator fault based on adaptive observer. *Transactions of the Institute of Measurement and Control*, 1–11.

Yang, T., Yi, X., Wu, J., Yuan, Y., Wu, D., Meng, Z., et al. (2019). A survey of distributed optimization. *Annual Reviews in Control*, *47*, 278–305.

Yang, H., Zhao, J., Lam, K., Xiong, Z., Wu, Q., & Xiao, L. (2022). Distributed deep reinforcement learning-based spectrum and power allocation for heterogeneous networks. *IEEE Transactions on Wireless Communication*, *21*(9), 6935–6948.

Yazdandoost Hamedani, E., & Aybat, N. S. (2017). Multi-agent constrained optimization of a strongly convex function. In *Global conference on signal and information processing* (pp. 558–562). IEEE.

Yemini, M., Nedić, A., Gil, S., & Goldsmith, A. J. (2022). Resilience to malicious activity in distributed optimization for cyberphysical systems. In *IEEE 61st conference on decision and control* (pp. 4185–4192). IEEE.

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

Zander, J. (1997). Radio resource management in future wireless networks: Requirements and limitations. *IEEE Communications Magazine*, *35*(8), 30–36.

Zargham, M., Ribeiro, A., Ozdaglar, A., & Jadbabaie, A. (2013). Accelerated dual descent for network flow optimization. *IEEE Transactions on Automatic Control*, *59*(4), 905–920.

Zayyani, H., Korki, M., & Marvasti, F. (2015). Dictionary learning for blind one bit compressed sensing. *IEEE Signal Processing Letters*, *23*(2), 187–191.

Zhang, Q., Cherkasova, L., & Smirni, E. (2007). A regression-based analytic model for dynamic resource provisioning of multi-tier applications. In *4th international conference on autonomic computing*. IEEE, 27–27.

Zhang, P., Li, Y., Kumar, N., Chen, N., Hsu, C., & Barnawi, A. (2022). Distributed deep reinforcement learning assisted resource allocation algorithm for space-air-ground integrated networks. *IEEE Transactions on Network and Service Management*, *20*(3), 3348–3358.

Zhang, J., You, K., & Cai, K. (2020). Distributed dual gradient tracking for resource allocation in unbalanced networks. *IEEE Transactions on Signal Processing*, *68*, 2186–2198.

Zholbaryssov, M., Hadjicostis, C., & Dominguez-Garcia, A. (2022). Fast coordination of distributed energy resources over time-varying communication networks. *IEEE Transactions on Automatic Control*, *68*(2), 1023–1038.

Zhou, Q., Shahidehpour, M., Paaso, A., Bahramirad, S., Alabdulwahab, A., & Abusorrah, A. (2020). Distributed control and communication strategies in networked microgrids. *IEEE Communications Surveys & Tutorials*, *22*(4), 2586–2633.

Zhu, Y., Ren, W., Yu, W., & Wen, G. (2019). Distributed resource allocation over directed graphs via continuous-time algorithms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *51*(2), 1097–1106.