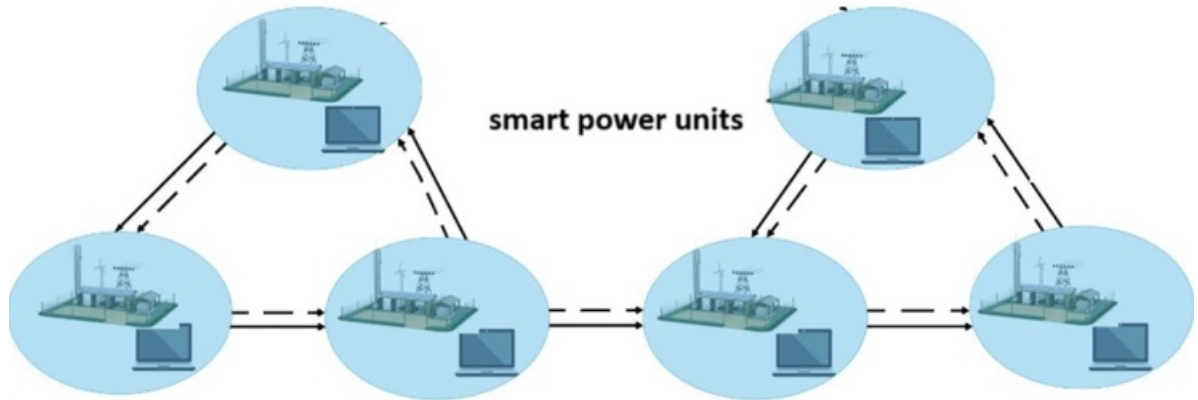


Graphical Abstract

Distributed Allocation and Resource Scheduling Algorithms Resilient to Link Failure

Mohammadreza Doostmohammadian, Sergio Pequito



Highlights

Distributed Allocation and Resource Scheduling Algorithms Resilient to Link Failure

Mohammadreza Doostmohammadian, Sergio Pequito

- Clearly defining the problem of distributed resource allocation as a formal mathematical optimization setup.
- Proposing a distributed resource allocation algorithm with optimal convergence over uniformly-connected networks while addressing all-time feasibility.
- Proving algorithm convergence and resilience under random link failure and time-delayed communication links

Distributed Allocation and Resource Scheduling Algorithms Resilient to Link Failure

Mohammadreza Doostmohammadian^a, Sergio Pequito^b

^a*Mechatronics Group Faculty of Mechanical Engineering Semnan University Semnan
Iran doost@semnan.ac.ir.*

^b*Department of Electrical and Computer Engineering and Institute for Systems and
Robotics Instituto Superior Tecnico University of Lisbon Portugal
sergio.pequito@tecnico.ulisboa.pt.*

Abstract

Distributed resource allocation (DRA) is fundamental to modern networked systems, spanning applications from economic dispatch in smart grids to CPU scheduling in data centers. Conventional DRA approaches require reliable communication, yet real-world networks frequently suffer from link failures, packet drops, and communication delays due to environmental conditions, network congestion, and security threats.

We introduce a novel resilient DRA algorithm that addresses these critical challenges, and our main contributions are as follows: (1) guaranteed constraint feasibility at all times, ensuring resource-demand balance even during algorithm termination or network disruption; (2) robust convergence despite sector-bound nonlinearities at nodes/links, accommodating practical constraints like quantization and saturation; and (3) optimal performance under merely uniformly-connected networks, eliminating the need for continuous connectivity.

Unlike existing approaches that require persistent network connectivity and provide only asymptotic feasibility, our graph-theoretic solution leverages network percolation theory to maintain performance during intermittent disconnections. This makes it particularly valuable for mobile multi-agent systems where nodes frequently move out of communication range. Theoretical analysis and simulations demonstrate that our algorithm converges to optimal solutions despite heterogeneous time delays and substantial link failures, significantly advancing the reliability of distributed resource allocation in practical network environments.

Keywords: Distributed resource allocation, graph theory, time-delay, percolation theory, convex optimization

1. Introduction

Modern networked systems increasingly rely on distributed algorithms for resource allocation, yet these systems face significant reliability challenges when operating in environments where communication disruptions are common. Resilient Distributed Resource Allocation (DRA) has emerged as a critical research area focused on maintaining optimal performance despite network imperfections such as link failures, packet drops, time delays, and cyber-attacks [1, 2]. Resilient DRA algorithms serve diverse applications, from CPU scheduling that optimally allocates computation loads among networked data centers [3, 4]; coordinated coverage control distributes surveillance tasks among mobile sensors to minimize distance-based costs [5, 6]; Task allocation for maximizing reliability of distributed computing systems [7, 8]; distributed energy management assigns power demand across generators to optimize generation costs [9, 10]; and electric vehicle charging coordination balances grid stability while meeting charging needs [11, 12].

Despite advances in resilient algorithms across various distributed domains, a critical gap remains in resource allocation systems that can maintain both constraint feasibility and optimality under realistic network conditions. Current DRA approaches – whether linear [13, 14, 15, 16], predefined-time [17, 18], attack-resilient [19], or event-triggered [20] – largely overlook the practical challenges of communication latency and nonlinear agent dynamics. Even approaches that address directed network topologies [21, 22, 23] remain vulnerable to link failures and depend on continuous network connectivity, an unrealistic assumption in many practical settings.

Communication disruptions are pervasive in real-world multi-agent systems through packet retransmission requirements, processing delays, and physical limitations. When DRA algorithms fail to address these realities, they risk violating resource-demand constraints and converging to suboptimal solutions, potentially triggering cascading instabilities or complete service disruptions. While consensus algorithms have made strides in addressing communication delays [24], existing delay-tolerant DRA approaches [25, 26] achieve constraint satisfaction only asymptotically and still depend on persistent network connectivity – assumptions that break down in environments with intermittent communication or mobile agents.

To bridge this critical gap, we introduce a graph-theoretic framework specifically designed to handle link failures and packet drops in dynamic network environments. Our approach recognizes that real-world networks frequently experience topology changes due to long-distance communication challenges, environmental interference, and node mobility. Unlike conventional algorithms that require continuous connectivity, our solution tolerates networks that are merely uniformly connected, meaning they may temporarily fragment but regain connectivity within bounded time intervals.

Furthermore, we develop a delay-tolerant mechanism that maintains feasibility and optimal convergence despite heterogeneous time delays across communication links. Most significantly, our resilient solution guarantees all-time feasibility – maintaining perfect balance between allocated resources and demand at every iteration. This critical property ensures that the algorithm can be safely terminated at any point without risking constraint violations, even during network disruptions or recovery phases.

In summary, our main contributions are as follows:

- (i) We introduce a distributed resource allocation algorithm that operates over uniformly-connected networks, significantly advancing resilience to link failures, packet drops, and time delays without requiring continuous network connectivity;

- (ii) Our algorithm achieves exact convergence to the optimizer despite sector-bound nonlinear mappings at both links and nodes, accommodating practical constraints such as log-scale quantization and clipping, a capability not previously addressed in distributed resource allocation literature;

- (iii) The proposed approach maintains perfect balance between allocated resources and demand at all iterations (all-time feasibility), even with nonlinear constraints on nodes/links. This critical property enables safe algorithm termination at any point without risking resource-demand imbalance or service disruption, a feature absent in dual-based formulations and ADMM-based methods [27, 28, 29, 30, 12]; and

- (iv) Our solution guarantees optimality, all-time feasibility, and convergence despite heterogeneous time delays and packet drops across the network. This comprehensive resilience ensures the algorithm functions effectively even when some links experience significant delays or complete failure, advancing beyond existing delay-tolerant approaches [25, 26] that cannot handle sector-bound nonlinearity while maintaining resource-demand feasibility.

The rest of the paper is organized as follows. Section 2 states the DRA problem and its applications. Section 3 presents the preliminaries on network

science and graph theory. Section 4 provides our main distributed algorithm to solve the DRA problem for both delay-free and delayed cases. Section 5 discusses the general convergence analysis, also in the presence of link failure and time delay. Section 6 presents the simulations and Section 7 concludes the paper.

2. The Resilient DRA Problem

Consider a multi-agent network where agents (i.e., nodes) have some (possibly nonlinear) dynamics and have to exchange information to find the solution to an optimization problem where some constraints need to be satisfied across the multi-agent network while the exchanged information is local. The network of agents is modelled as a graph of n nodes, denoted by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ that might be time-varying, i.e., for a fixed set of nodes $\mathcal{V}$ the link set $\mathcal{E}$ may change over time. Its associated adjacency matrix, denoted by W , is symmetric and entry $W_{ij} > 0$ denotes the weighting factor on the information shared over link $(j, i) \in \mathcal{E}$. $W_{ij} = 0$ implies that $(j, i) \notin \mathcal{E}$. Define $\mathcal{N}_i = \{j | (j, i) \in \mathcal{E}\}$ as the set of neighbours of the node/agent i for data-sharing.

The DRA problem in its most general representation is in the form of coupling-constraint distributed optimization. The objective function represents the sum of local cost (or loss) at agents, each having access to a certain amount of resources z_i . The weighted sum of the shared resources is constant b . The problem is to minimize the objective by assigning the proper amount of resources to the agents subject to the resource-demand equality constraint as follows

$$\begin{aligned} \min_{\mathbf{z}} \quad & H(\mathbf{z}) = \sum_{i=1}^n h_i(z_i), \\ \text{s.t.} \quad & \sum_{i=1}^n a_i z_i = b. \end{aligned} \tag{1}$$

By a simple change of variable $a_i z_i = x_i$, the problem changes into the following standard form

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

The function $f_i : \mathbb{R} \mapsto \mathbb{R}$ is strictly convex and $2u$ -smooth, i.e., it satisfies $0 < \frac{d^2 f_i}{dx_i^2} < 2u$. In many applications, the constraint $\sum_{i=1}^n x_i = b$ is required to be held at all iterations under the distributed solution. This is referred to as *all-time feasibility* and ensures that the balance between allocated states $\sum_{i=1}^n x_i$ and demand b holds at any termination time of the algorithm with no service disruption.

The states of the agents might be subject to local box constraints which represent simple upper-bound and lower-bound on the states in the form

$$\begin{aligned} \min_{\mathbf{x}} \quad & F(\mathbf{x}) = \sum_{i=1}^n f_i(x_i), \\ \text{s.t.} \quad & \sum_{i=1}^n x_i = b, \\ & m_i \leq x_i \leq M_i. \end{aligned} \tag{3}$$

The local constraints in (3) can be efficiently addressed by incorporating barrier functions or additive penalty terms into the objective function. This approach introduces a controllable optimality gap as discussed in [31, 32].

A common penalty formulation is $f^\sigma(x_i) = \sigma([x_i - M_i]^+ + [m_i - x_i]^+)$, where $[u]^+ = (\max\{0, u\})^m$ with $m \in \mathbb{N}_{\geq 2}$. The parameters m and σ should be sufficiently large to ensure adequate penalization of constraint violations, thereby guiding the solution toward the feasible region.

For applications requiring smoother penalty transitions, the logarithmic approximation $f^\mu(u) = \frac{1}{\mu} \log(1 + \exp(\mu u))$ offers an attractive alternative, where u represents either $[x_i - M_i]$ or $[m_i - x_i]$. This formulation converges to $\max\{u, 0\}$ as μ increases, providing arbitrarily close approximations to the exact constraint with appropriate parameter selection. See [31] for a comprehensive analysis of these penalty methods and their convergence properties.

To facilitate reading and ease the exposition that follows, all required notation is summarized in Table 1.

2.1. Resilient DRA Problem Formulation

Hereafter, we address the challenge of distributed resource allocation over unreliable networks characterized by communication disruptions. Unlike conventional approaches that assume reliable communication, we explicitly model and mitigate the effects of random packet drops and link failures that occur in practical networked systems.

Table 1: List of symbols

$\mathbf{x}, \mathbf{z}$	column state vectors
$\mathbf{1}_n, \mathbf{0}_n$	column vector of ones and zeros of size n
$\mathbf{x}^*$	optimal state
$\mathcal{S}_b$	feasibility set (associated with b)
x_i	state of node i
k	discrete time index
$F(\cdot)$	global cost function
$\mathcal{G}, \mathcal{V}, \mathcal{E}$	network, set of nodes, set of links
$\nabla F(\cdot)$	gradient of $F(\cdot)$
d_G	network diameter
$f_i(\cdot)$	local cost function
W, L	weight matrix, Laplacian matrix
$\partial f_i(\cdot)$	first derivative of $f_i(\cdot)$
λ_i	i th eigenvalue of L
τ_{ij}	time-delay of link (i, j)
$\mathcal{N}_i$	neighbors of node i
$\bar{\tau}$	global upper-bound on the time-delay
g_n, g_l	nonlinear mapping at the nodes/links
κ_n, κ_l	lower sector-bound on g_n, g_l
$\mathcal{K}_n, \mathcal{K}_l$	upper sector-bound on g_n, g_l
u	upper bound on second gradient of f_i
$\mathcal{I}(\cdot)$	indicator function
n	number of agents/nodes
ζ	dispersion parameter vector
p_c, p_l	percolation threshold, link failure probability

Our focus on unreliable networks is motivated by several key considerations:

1. **Random Structure of Real Networks:** Many real-world networks exhibit inherently random characteristics [33]. By modelling networks as random structures, we can better analyze their interconnection properties and behaviour under varying conditions.
2. **Unpredictable Link Failures:** In practical deployments, link failures occur unpredictably due to hardware malfunctions, environmental interference, or congestion. Our random network model allows rigorous evaluation of algorithm resilience against such failures.
3. **Scalability Analysis:** Random network models facilitate scalability analysis as system complexity increases, providing insights into how distributed algorithms perform with expanding network size and connectivity changes.

Our resilient DRA formulation extends the standard problem to incorporate:

1. **Time-Varying Connectivity:** We model $\mathcal{G}(k) = \{\mathcal{V}, \mathcal{E}(k)\}$ as a dynamic graph where the edge set $\mathcal{E}(k)$ changes over time due to communication disruptions.
2. **Probabilistic Link Failures:** Each communication link $(i, j) \in \mathcal{E}$ has a probability p_l of failure at any time step, representing packet drops, congestion, or physical disconnection.
3. **Nonlinear Communication Effects:** We account for practical nonlinearities in information exchange through sector-bound mappings at both nodes and links $g_n : \mathbb{R} \mapsto \mathbb{R}$, and $g_l : \mathbb{R} \mapsto \mathbb{R}$, where $\kappa_n z < g_n(z) < K_n z$ and $\kappa_l z < g_l(z) < K_l z$ represent the sector bounds on these nonlinear mappings.
4. **Heterogeneous Time Delays:** Communication between agents may experience varying delays $\tau_{ij}(k) \leq \bar{\tau}$ that are time-varying, random, and potentially different for each link. These delays are incorporated into our agent dynamics as

$$x_i(k+1) = x_i(k) - \eta_\tau \sum_{j \in \mathcal{N}_i} \sum_{r=0}^{\bar{\tau}} W_{ij} g_n(g_l(\partial f_i(k-r)) - g_l(\partial f_j(k-r))) I_{k-r,ij}(r), \quad (4)$$

where $I_{k-r,ij}(r)$ is an indicator function defined as

$$I_{k,ij}(\tau) = \begin{cases} 1, & \text{if } \tau_{ij}(k) = \tau, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

This comprehensive model captures the realistic challenges of distributed resource allocation in practice, where network disruptions can occur due to hardware malfunctions, interference, congestion, or buffer overflow. Our solution approach, based on the percolation theory discussed in Section 3.2, provides rigorous guarantees for algorithm performance despite these communication challenges.

3. Preliminaries on Network Science

This section establishes the mathematical and theoretical foundations necessary for our resilient distributed resource allocation approach. We begin by introducing essential concepts from algebraic graph theory in Section 3.1,

where we examine the properties of Laplacian matrices and their eigenspectra. Next, in Section 3.2, we introduce network percolation theory, which provides the analytical framework for characterizing network resilience under random link failures. These mathematical preliminaries form the essential basis for both our algorithm design in Section 4 and the rigorous convergence analysis presented in Section 5, particularly when proving algorithm resilience to link failures and communication delays.

3.1. Algebraic Graph Theory

Next, we present essential graph-theoretic concepts relevant to our analysis. Given a multi-agent network with adjacency matrix W , we define the diagonal degree matrix $D := \text{diag}(\sum_{j=1}^n W_{ij})$ and the corresponding Laplacian matrix $L = D - W$. For a connected graph with symmetric L , the spectrum of the Laplacian contains exactly one zero eigenvalue, with $\mathbf{1}_n$ (the all-ones vector) as its associated eigenvector. When the network becomes disconnected or fragmented into islands, the Laplacian matrix exhibits multiple zero eigenvalues, with the number corresponding to the count of connected components. The spectral properties of the Laplacian matrix L fundamentally characterize the consensus behaviour and information diffusion across the network $\mathcal{G}$. The following lemma, derived from the Courant-Fischer theorem, provides crucial insight into quantifying consensus *disagreement* in multi-agent networks [34].

Lemma 1. [34] *Given a symmetric Laplacian L and vector $\mathbf{x} \in \mathbb{R}^n$, define $\bar{\mathbf{x}} =: \mathbf{x} - \frac{\mathbf{1}_n^\top \mathbf{x}}{n} \mathbf{1}_n$ as the dispersion state. Then, the following hold:*

$$\mathbf{x}^\top L \mathbf{x} = \bar{\mathbf{x}}^\top L \bar{\mathbf{x}}, \quad (6)$$

$$\lambda_2 \|\bar{\mathbf{x}}\|_2^2 \leq \mathbf{x}^\top L \mathbf{x} \leq \lambda_n \|\bar{\mathbf{x}}\|_2^2, \quad (7)$$

where λ_n and λ_2 respectively denote the largest and smallest non-zero eigenvalue of L .

One can extend the results of Lemma 1 to consider sector-bound nonlinearity at the nodes/links as follows.

Corollary 1. *Consider two vector states $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ where the entries $y_i = g(x_i)$ for $i \in \{1, \dots, n\}$ such that the sector-bound mapping $g(\cdot)$ is sign-preserving, odd, and monotonically non-decreasing implying $0 < \kappa < \frac{y_i}{x_i} < \mathcal{K}$,*

where $\kappa, \mathcal{K} \in \mathbb{R}^+$ are lower/upper sector bound. Let $\bar{\mathbf{x}} =: \mathbf{x} - \frac{\mathbf{1}_n^\top \mathbf{x}}{n} \mathbf{1}_n$ and $\bar{\mathbf{y}} =: \mathbf{y} - \frac{\mathbf{1}_n^\top \mathbf{y}}{n} \mathbf{1}_n$. Then, the following holds:

$$\mathbf{x}^\top L \mathbf{y} = \bar{\mathbf{x}}^\top L \bar{\mathbf{y}}, \quad (8)$$

$$\lambda_2 \bar{\mathbf{x}}^\top \bar{\mathbf{y}} \leq \mathbf{x}^\top L \mathbf{y} \leq \lambda_n \bar{\mathbf{x}}^\top \bar{\mathbf{y}}. \quad (9)$$

Proof. Recall that for symmetric W matrix, its associated Laplacian L satisfies $\mathbf{1}_n^\top L = L \mathbf{1}_n = \mathbf{0}_n$. Then,

$$\begin{aligned} \mathbf{x}^\top L \mathbf{y} &= (\bar{\mathbf{x}} + \frac{\mathbf{1}_n^\top \mathbf{x}}{n} \mathbf{1}_n)^\top L (\bar{\mathbf{y}} + \frac{\mathbf{1}_n^\top \mathbf{y}}{n} \mathbf{1}_n) \\ &= \bar{\mathbf{x}}^\top L \bar{\mathbf{y}}, \end{aligned}$$

which proves (8). The proof of the second part follows from a special case of Courant-Fischer's theorem in [35]. Denote the eigenvalues of L as $\lambda_1 < \lambda_2 \leq \lambda_3 \leq \dots \leq \lambda_n$ with $\lambda_1 = 0$ associated with the eigenvector $\mathbf{1}_n$. We can express $\mathbf{x}, \mathbf{y}$ in terms of the eigenbasis of L , denoted by unit vectors $v_1, \dots, v_n$, as

$$\mathbf{x} = \alpha_1 v_1 + \dots + \alpha_n v_n, \text{ and } \mathbf{y} = \beta_1 v_1 + \dots + \beta_n v_n,$$

where $0 < \kappa < \frac{\alpha_i}{\beta_i} < \mathcal{K}$. Expanding $\mathbf{x}^\top L \mathbf{y} = \bar{\mathbf{x}}^\top L \bar{\mathbf{y}}$, and using the eigendecomposition of L , we get

$$\bar{\mathbf{x}}^\top L \bar{\mathbf{y}} = \alpha_1 \beta_1 \lambda_1 + \alpha_2 \beta_2 \lambda_2 + \dots + \alpha_n \beta_n \lambda_n,$$

where due to sector-bound condition and monotonicity, we have $0 < \kappa \beta_i^2 < \alpha_i \beta_i < \mathcal{K} \beta_i^2$. We also have $\bar{\mathbf{x}}^\top \bar{\mathbf{y}} = \alpha_1 \beta_1 + \dots + \alpha_n \beta_n$. Then, following the non-decreasing order of eigenvalues and the fact that $\lambda_1 = 0$ is associated with the eigenvector $\mathbf{1}_n$, we have

$$\lambda_2 = \min_{\bar{\mathbf{x}}, \bar{\mathbf{y}} \neq 0, \bar{\mathbf{x}} \perp \mathbf{1}_n, \bar{\mathbf{y}} \perp \mathbf{1}_n} \frac{\bar{\mathbf{x}}^\top L \bar{\mathbf{y}}}{\bar{\mathbf{x}}^\top \bar{\mathbf{y}}}, \text{ and } \lambda_n = \max_{\bar{\mathbf{x}}, \bar{\mathbf{y}} \neq 0, \bar{\mathbf{x}} \perp \mathbf{1}_n, \bar{\mathbf{y}} \perp \mathbf{1}_n} \frac{\bar{\mathbf{x}}^\top L \bar{\mathbf{y}}}{\bar{\mathbf{x}}^\top \bar{\mathbf{y}}}$$

that is known as the Rayleigh quotient [35], from which it readily follows (9). $\square$

Lemma 2. Let $\mathbf{y} = g(\mathbf{x})$ such that the sector-bound mapping $g(\cdot)$ is sign-preserving, odd, and monotonically non-decreasing such that the element-wise relations $\kappa x_i < y_i < \mathcal{K} x_i$ hold for all i, j . Then, it follows that

$$\lambda_2 \kappa \|\bar{\mathbf{x}}\|_2^2 \leq \mathbf{x}^\top L \mathbf{y} \leq \lambda_n \mathcal{K} \|\bar{\mathbf{x}}\|_2^2. \quad (10)$$

Proof. From the definition of $\bar{\mathbf{x}}, \bar{\mathbf{y}}$, we have

$$\begin{aligned}
\bar{\mathbf{x}}^\top \bar{\mathbf{y}} &= (\mathbf{x} - \frac{\mathbf{1}_n^\top \mathbf{x}}{n} \mathbf{1}_n)^\top (\mathbf{y} - \frac{\mathbf{1}_n^\top \mathbf{y}}{n} \mathbf{1}_n) \\
&= \mathbf{x}^\top \mathbf{y} - \frac{\mathbf{1}_n^\top \mathbf{x}}{n} \mathbf{1}_n^\top \mathbf{y} - \frac{\mathbf{1}_n^\top \mathbf{y}}{n} \mathbf{1}_n^\top \mathbf{x} + \frac{\mathbf{1}_n^\top \mathbf{y} \mathbf{1}_n^\top \mathbf{x}}{n} \mathbf{1}_n^\top \mathbf{1}_n \\
&= \mathbf{x}^\top \mathbf{y} + \frac{\mathbf{1}_n^\top \mathbf{y} \mathbf{1}_n^\top \mathbf{x}}{n} (n - 2).
\end{aligned} \tag{11}$$

From the sector-bound relation, we have $\kappa \mathbf{x} < \mathbf{y} < \mathcal{K} \mathbf{x}$ and applying this to (11), we obtain

$$\begin{aligned}
\kappa(\mathbf{x}^\top \mathbf{x} + \frac{\mathbf{1}_n^\top \mathbf{x} \mathbf{1}_n^\top \mathbf{x}}{n} (n - 2)) &\leq \mathbf{x}^\top \mathbf{y} + \frac{\mathbf{1}_n^\top \mathbf{y} \mathbf{1}_n^\top \mathbf{x}}{n} (n - 2) \\
&\leq \mathcal{K}(\mathbf{x}^\top \mathbf{x} + \frac{\mathbf{1}_n^\top \mathbf{x} \mathbf{1}_n^\top \mathbf{x}}{n} (n - 2)).
\end{aligned} \tag{12}$$

Similar to (11), from the definition of $\bar{\mathbf{x}}$, it readily follows that

$$\mathbf{x}^\top \mathbf{x} + \frac{\mathbf{1}_n^\top \mathbf{x} \mathbf{1}_n^\top \mathbf{x}}{n} (n - 2) = \bar{\mathbf{x}}^\top \bar{\mathbf{x}} = \|\bar{\mathbf{x}}\|_2^2. \tag{13}$$

Combining Eqs. (11), (12), and (13), we get $\kappa \|\bar{\mathbf{x}}\|_2^2 \leq \bar{\mathbf{x}}^\top \bar{\mathbf{y}} \leq \mathcal{K} \|\bar{\mathbf{x}}\|_2^2$. Then, combining this with Eq. (9) in Corollary 1, we obtain (10). $\square$

Note that the second smallest eigenvalue of L , denoted by λ_2 , is known as algebraic connectivity. It is known from the literature that adding links to the network $\mathcal{G}$ increases its algebraic connectivity [34]. These results can be extended to the union network as follows. First, let us recall the definition of the uniformly-connectivity.

Definition 1. *For a uniformly-connected network, there exists bounded time $T > 0$ such that $\mathcal{G}_T(k) = \cup_k^{k+T} \mathcal{G}(k)$ is connected for all k .*

Remark 1. *The boundedness of T in Definition 1 is essential for uniform-connectivity, as it guarantees the network regains connectivity within a finite time window. Consider a counterexample: a network that connects only during sporadic intervals and remains disconnected throughout most of the time domain. In such a case, no bounded T exists for which $\mathcal{G}_T(k) = \cup_{j=k}^{k+T} \mathcal{G}(j)$ is connected for all $k \geq 0$. Therefore, this network fails to satisfy the uniform-connectivity property. Uniform connectivity requires not just eventual reconnection, but reconnection within a consistent, bounded timeframe across the entire time domain.* $\circ$

One can extend the results of Lemma 1 and Corollary 1 to uniformly-connected networks as follows.

Corollary 2. *Consider uniformly-connected network $\mathcal{G}_T(k) = \cup_k^{k+T} \mathcal{G}(k)$ with associated Laplacian L_T and eigenvalues λ_{2T} and λ_{nT} . Then, $\lambda_{2T} \bar{\mathbf{x}}^\top \bar{\mathbf{y}} \leq \mathbf{x}^\top L_T \mathbf{y} \leq \lambda_{nT} \bar{\mathbf{x}}^\top \bar{\mathbf{y}}$ and, following from Lemma 2, we have $\lambda_{2T} \kappa \|\bar{\mathbf{x}}\|_2^2 \leq \mathbf{x}^\top L_T \mathbf{y} \leq \lambda_{nT} \kappa \|\bar{\mathbf{x}}\|_2^2$.*

3.2. Percolation Theory

Percolation theory is a mathematical framework that studies the behaviour of connected clusters in random graphs or networks. Specifically, the theory investigates how global connectivity emerges from local random processes, focusing on phase transitions where the system's properties change dramatically at critical thresholds. The key challenge percolation theory addresses is determining when a network maintains its essential connectivity properties despite random removal of links or nodes[36, 37]. In many control applications, the connectivity of the graph plays a key role in the convergence. This is addressed by the notion of *bond-percolation* as defined next.

Definition 2. [38] *Let denote the probability of link removal by p_l . Then, bond-percolation of the network refers to the probability threshold p_c (or the likelihood) that a network $\mathcal{G}$ will lose its connectivity when links are randomly removed or percolated from $\mathcal{G}$. In other words, for $p_l > p_c$ there is no giant connected component spanning all nodes $\mathcal{V}$ with probability 1.*

In this context, Kolmogorov's zero-one law states a two-sided phase transition point in network connectivity, implying that there is a sharp, binary phase transition in terms of network connectivity as the percolation probability changes [39]. This means that there is a critical threshold value of $p_c \in [0, 1]$, where the nature of network connectivity changes considerably. For $p_l < p_c$ there is a subcritical regime, where a connected component spanning the entire network exists with probability 1. The supercritical regime occurs for $p_l > p_c$, implying that the network is fragmented, and there is no connected path spanning from one side of the network to the other [40], i.e., the probability of network connectivity is zero with certainty.

This is an essential concept for understanding how network properties change as bonds/links are randomly added or removed. For example, given an Erdos-Renyi (ER) graph [41] of n nodes and linking probability $p \in [0, 1]$

its bond-percolation threshold is equal to

$$p_c = 1 - \frac{1}{\overline{d_G}}, \quad (14)$$

with $\overline{d_G} = \frac{(n-1)p}{2}$ as the average node degree [42, 43].

Properties of ER random graphs hold asymptotically. This means that an ER random graph can become *asymptotically almost surely connected* for a given threshold but when examined for a small graph with the same probability, it is disconnected.

It is important to note that the connectivity properties of ER random graphs are established in the asymptotic limit. Therefore, throughout the remainder of this paper, when we refer to connectivity for ER random graphs, we specifically mean connectivity in the *asymptotically almost surely (a.a.s.)* sense. This distinction is crucial as it captures the probabilistic nature of connectivity in large networks. Furthermore, these random graph models provide powerful analytical tools for estimating percolation thresholds in practical applications, enabling us to predict and enhance the resilience of large-scale, complex networked control systems against random failures or attacks.

4. The Proposed DRA Algorithm

This section presents our distributed resource allocation (DRA) algorithm designed to address the challenges of link failures and time delays in multi-agent networks. In Section 4.1, we introduce the fundamental dynamics for delay-free networks and analyze its key properties, including all-time feasibility and equilibrium conditions. In Section 4.2, we extend the algorithm to handle heterogeneous time delays while preserving these essential properties. Finally, in Section 4.3, we summarize the complete algorithm and highlight its advantages compared to existing approaches in the literature.

4.1. Solution to the Delay-Free Distributed Dynamics

Consider a group of n networked agents which update their state $x_i(k) \in \mathbb{R}$ (state of agent i at time step k) satisfying the following dynamics:

$$x_i(k+1) = x_i(k) - \eta \sum_{j \in \mathcal{N}_i} W_{ij} g_n \left(g_l(\partial f_i(k)) - g_l(\partial f_j(k)) \right), \quad (15)$$

with $\partial f_i = \frac{df_i}{dx_i}$, and $\eta > 0$ as step-rate. The nonlinear mappings $g_n : \mathbb{R} \mapsto \mathbb{R}$ and $g_l : \mathbb{R} \mapsto \mathbb{R}$ denote possible nonlinearity at the nodes and the links, respectively. These mappings satisfy the following properties.

Assumption 1. *The nonlinear functions g_n and g_l are odd and sign-preserving, and also satisfy the sector bound condition in the form $\kappa_n z < g_n(z) < \mathcal{K}_n z$ and $\kappa_l z < g_l(z) < \mathcal{K}_l z$.*

Examples of such nonlinear mappings in real-world applications are, just to mention a few, the log-scale quantization given by

$$g(z) = \text{sgn}(z) \exp \left(\rho \left[\frac{\log(|z|)}{\rho} \right] \right),$$

as well as the bounded-domain saturation (clipping) defined as

$$g(z) = \begin{cases} R \text{sgn}(z) & |z| > R \\ z & |z| \leq R \end{cases}$$

with saturation level R . For these nonlinear functions the upper/lower sector bound parameters are readily obtainable; for example, for log-scale quantization we have $\kappa = 1 - \frac{\rho}{2}$, $\mathcal{K} = 1 + \frac{\rho}{2}$ [44, 45] and for bounded-domain saturation, given the domain $|z| \leq D_{max}$ and $D_{max} > R$, we have $\kappa = \frac{R}{D_{max}}$ and $\mathcal{K} = 1$. Some works in the literature even apply signum-based functions to improve the convergence rate [46].

Notice that the dynamics in (15) satisfies the *Laplacian-tracking* property, i.e., its summation satisfies the following:

$$\kappa_n \kappa_l L \nabla F(k) \leq \sum_{j \in \mathcal{N}_i} W_{ij} g_n \left(g_l(\partial f_i(k)) - g_l(\partial f_j(k)) \right) \leq \mathcal{K}_n \mathcal{K}_l L \nabla F(k). \quad (16)$$

Additionally, we have the following assumption regarding the agents' exchange of information.

Assumption 2. *The agents communicate over an undirected network $\mathcal{G}(k) = \mathcal{V}, \mathcal{E}(k)$ with a symmetric weight matrix $W(k) = [W_{ij}(k)]_{i,j \in \mathcal{V}}$ which is (possibly) time-varying and uniformly-connected.*

Assumption 2 serves two critical purposes: First, it guarantees that the proposed dynamics maintains all-time feasibility (also called anytime feasibility), thereby satisfying the equality constraint at every iteration as formally

demonstrated in Lemma 3. Second, the uniform connectivity condition is essential for ensuring algorithmic resilience against communication imperfections such as time-delays and packet drops, which will be rigorously analyzed in subsequent sections of this paper.

Before proceeding with the analysis of the algorithm's feasibility, we introduce the set of vectors that satisfy our equality constraint, which will be essential for characterizing the feasible region of our distributed resource allocation problem.

Definition 3. Define $\mathcal{S}_b$ as the set of all feasible points satisfying the equality constraint $\{\mathbf{x} \in \mathbb{R}^n | \mathbf{1}_n^\top \mathbf{x} = b\}$ with $\mathbf{1}_n$ as all-one vector of size n .

Having established the feasible set $\mathcal{S}_b$ in Definition 3, we now demonstrate that our proposed algorithm ensures the solution trajectory remains within this set throughout its execution¹. The following lemma proves that the equality constraint $\sum_{i=1}^n x_i = b$ (or $\mathbf{1}_n^\top \mathbf{x} = b$) in (2) is satisfied at all iterations of the algorithm, a property we refer to as *all-time feasibility*.

Lemma 3. Given $g_n(\cdot)$ and $g_l(\cdot)$ satisfying Assumption 1 and $\mathbf{x}(0) \in \mathcal{S}_b$, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ satisfying Assumption 2, the solution under dynamics (15) is all-time feasible and the equality constraints is satisfied at all times, i.e., $\mathbf{x}(k) \in \mathcal{S}_b$ for $k > 0$.

Proof. At any time $k > 0$, we have

$$\mathbf{1}_n^\top \mathbf{x}(k+1) = \mathbf{1}_n^\top \mathbf{x}(k) - \eta \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} W_{ij} g_n \left(g_l(\partial f_i(k)) - g_l(\partial f_j(k)) \right). \quad (17)$$

We have the functions $g_n(\cdot)$ and $g_l(\cdot)$ as sign-preserving odd mappings and we also have $W_{ij} = W_{ji}$. This implies that the dual summation in (17) is equal to zero and $\mathbf{1}_n^\top \mathbf{x}(k+1) = \mathbf{1}_n^\top \mathbf{x}(k)$. Recalling that $\mathbf{x}(0) \in \mathcal{S}_b$ this implies that $\mathbf{x}(k) \in \mathcal{S}_b$ for all $k > 0$. $\square$

This critical all-time feasibility property guarantees that the system can be terminated at any point while preserving the feasibility of the equality

¹For initialization, we need $\sum_{i=1}^n x_i(0) = b$. There exist distributed algorithms to satisfy this initial constraint and locally allocate the initial values, see [47, Algorithm 2] for example.

constraint. In practical distributed applications such as economic dispatch and generator coordination, this property is particularly valuable as it ensures continuous service without disruption – the power demand-supply balance is preserved throughout the algorithm’s execution regardless of when termination occurs.

Furthermore, we can establish that the equilibrium $\mathbf{x}^*$ of the proposed dynamics (15) must satisfy $\nabla F(\mathbf{x}^*) \in \text{span}(\mathbf{1}_n)$. This condition has important implications: any point where $\nabla F(\mathbf{x}^*) \notin \text{span}(\mathbf{1}_n)$ cannot serve as an equilibrium of our system (15). The formal proof of this characteristic is presented in the following lemma.

Lemma 4. *Under the uniform-connectivity of undirected multi-agent network $\mathcal{G}$, the equilibrium $\mathbf{x}^*$ of the dynamics (15) is in the form $\nabla F(\mathbf{x}^*) \in \text{span}(\mathbf{1}_n)$.*

Proof. We prove this by contradiction. Suppose that $\hat{\mathbf{x}}$ is the equilibrium of (15), i.e., $\hat{x}_i(k+1) = \hat{x}_i(k)$ for $\forall i$, for which $\nabla F(\hat{\mathbf{x}}) \notin \text{span}(\mathbf{1}_n)$. This implies that there exists (at least) two agents a and b such that $\partial f_a(\hat{x}_a) \neq \partial f_b(\hat{x}_b)$. Following the definition of connectivity, there exists a path over the connected (union) network $\mathcal{G}_T$ between these two agents a, b . Then, for two neighboring agents $j \in \mathcal{N}_i$ on this path we have $\partial f_i(\hat{x}_i) \neq \partial f_j(\hat{x}_j)$. Therefore, invoking (15), we have

$$\sum_{j \in \mathcal{N}_i} W_{ij} g_n \left(g_l(\partial f_i(k)) - g_l(\partial f_j(k)) \right) \neq 0, \quad (18)$$

and, therefore, $\hat{x}_i(k+1) \neq \hat{x}_i(k)$. This violates the definition of equilibrium for $\hat{\mathbf{x}}$, and the conclusion follows. $\square$

4.2. Solution in the Presence of Network Delays

In the previous subsection, we presented a delay-free distributed algorithm that possesses the desirable properties of all-time feasibility and optimal convergence under uniform network connectivity. However, real-world communication networks often experience unavoidable time delays due to various factors such as packet retransmission, processing time, congestion, and physical constraints. These delays can significantly impact algorithm performance and potentially compromise both the feasibility and optimality of traditional distributed solutions.

We now extend our approach to address heterogeneous time delays while preserving the key properties of our delay-free algorithm. This extension is crucial for practical implementations, as it ensures the resilience of our distributed resource allocation method in realistic networking conditions. The following assumption formalizes the delay model we consider in the sequel.

Assumption 3. *The time delays are, in general, heterogeneous at different links and at different time instants $k \in \mathbb{Z}^+$. The time delay at a bidirectional link (i, j) where $i, j \in \{1, 2, \dots, n\}$ is denoted by $\tau_{ij}(k) \in \{0, 1, 2, \dots, \bar{\tau}\} \leq \bar{\tau}$ and, in the most general form, is arbitrary, random, and bounded by a maximum possible delay $\bar{\tau} \in \mathbb{Z}^+$.*

Additionally, we assume the information sent from the sender node i to recipient node j is *time-stamped* and, thus, the delay is known to the node j .

Therefore, in the presence of network delays, the proposed agents' update should satisfy the following dynamics:

$$x_i(k+1) = x_i(k) - \eta_\tau \sum_{j \in \mathcal{N}_i} \sum_{r=0}^{\bar{\tau}} W_{ij} g_n \left(g_l(\partial f_i(k-r)) - g_l(\partial f_j(k-r)) \right) \mathcal{I}_{k-r,ij}(r), \quad (19)$$

with η_τ as the step rate in the presence of time-delays. Besides, to capture the presence and timing of delays in our proposed solution, we employ an indicator function $\mathcal{I}$ defined as

$$\mathcal{I}_{k,ij}(\tau) = \begin{cases} 1, & \text{if } \tau_{ij}(k) = \tau, \\ 0, & \text{otherwise.} \end{cases} \quad (20)$$

In the worst case of unknown asymmetric time delays, both agents i, j can update their mutual information based on the maximum delay at every link or the maximum possible delay $\bar{\tau}$, i.e., $\mathcal{I}_{k,ij}(\bar{\tau}) = \mathcal{I}_{k,ji}(\bar{\tau}) = 1$. Later, we will also make explicit that η_τ depends on the time-delay bound $\bar{\tau}$.

Using the same arguments as in Lemma 3 and 4, we can establish that both the all-time feasibility and equilibrium properties extend naturally to the time-delayed dynamics (19). This connection arises from a fundamental insight: *delayed information exchange effectively manifests as temporal changes in the network topology.*

Consider a concrete example: when information exchange between nodes i and j experiences a delay from time k to time $k + \tau$, we can conceptualize

this as removing link (i, j) from $\mathcal{G}(k)$ and adding it to $\mathcal{G}(k + \tau)$ instead. This topological reframing allows us to leverage our existing analysis framework.

By recasting time delays as topology modifications, we can directly extend the properties established for the original dynamics (15) to the delay-inclusive dynamics (19). The mathematical structure remains consistent, with the primary difference being the temporal distribution of connections.

Furthermore, since all delays are bounded by $\bar{\tau}$, we can guarantee that the effective topology associated with the delayed dynamics maintains uniform connectivity over a slightly extended time window. Specifically, the union network $\mathcal{G}_{T+\bar{\tau}}(k) = \cup_k^{k+T+\bar{\tau}} \mathcal{G}(k)$ remains connected, preserving the essential connectivity property required for convergence, albeit over a longer time horizon that accommodates the maximum possible delay.

This topological perspective on delays provides a unifying framework for analyzing the resilience of our algorithm without requiring fundamentally different mathematical machinery for the delayed case.

4.3. The Complete Resilient DRA Algorithm

Our proposed distributed allocation and scheduling algorithm is presented in Algorithm 1. The method follows a structured approach with two key phases per iteration:

First, each node exchanges gradient information derived from its local objective function with neighboring nodes across the network $\mathcal{G}$. This information sharing forms the foundation of our distributed approach, enabling coordination without centralized control.

Second, nodes update their states according to the resilient DRA dynamics in (19), which explicitly accounts for heterogeneous time delays across communication links. This update mechanism iteratively guides the system toward the optimal solution of the constrained optimization problem while maintaining all-time feasibility.

Remark 2. *For the special case where no communication delays are present (i.e., $\tau_{ij}(k) = 0$ for all links), our algorithm naturally simplifies to the delay-free dynamics in (15), demonstrating the unified nature of our theoretical framework. This adaptive behaviour allows the algorithm to seamlessly operate in both ideal and delay-prone network environments.*

Remark 3. *In the absence of node nonlinearity (i.e., linear mapping $g_n(u) = u$), the algorithm converges over general weight-balanced directed networks.*

Algorithm 1: The Resilient DRA Algorithm.

1 **Given:** Uniformly connected network $\mathcal{G}$, W , η , cost $f_i(\cdot)$
2 **Initialization:** $k = 0$, $\mathbf{x}(0) \in \mathcal{S}_b$
3 **while** *termination criteria NOT hold* **do**
4 Each node i receives gradient data from $j \in \mathcal{N}_i$;
5 Each node i updates $x_i(k+1)$ via Eq. (19);
6 Each node i shares its gradient information with its neighbors
 $j \in \mathcal{N}_i$;
7 $k \leftarrow k + 1$;
8 **Return:** Assigned resources $\mathbf{x}(k)$ and associated cost $F(\mathbf{x}(k))$;

To complete our algorithmic analysis, we now establish the computational requirements of our approach through a formal complexity characterization.

Proposition 1 (Computational Complexity). *The computational complexity of the proposed DRA algorithm is $\mathcal{O}(kn^2)$ for k iterations in a network of n agents when g_n and g_l are linear functions. When g_n and g_l are nonlinear, the complexity becomes $\mathcal{O}(kn^2 + kn \cdot C(g))$, where $C(g)$ denotes the computational cost of evaluating the nonlinear functions.*

Proof. We analyze the computational complexity by examining the operations performed at each iteration of Algorithm 1 across all nodes:

1. Local gradient computation: Each agent i computes $\partial f_i(x_i(k))$, requiring $\mathcal{O}(1)$ operations per agent, for a total of $\mathcal{O}(n)$ operations across the network.
2. Information exchange: In the worst case, the network is fully connected, and each agent communicates with all other agents. This requires $\mathcal{O}(n)$ message transmissions per agent, for a total of $\mathcal{O}(n^2)$ communication operations.
3. State update: For each agent i , the state update in Equation (15) or (19) requires:
 - When g_n and g_l are linear: Computing the weighted sum of gradient differences across all neighbors requires $\mathcal{O}(|\mathcal{N}_i|)$ operations, where $|\mathcal{N}_i|$ is the number of neighbors of agent i . In the worst

case, $|\mathcal{N}_i| = n - 1$, leading to $\mathcal{O}(n)$ operations per agent and $\mathcal{O}(n^2)$ operations across all agents.

- When g_n and g_l are nonlinear: Each evaluation of g_n or g_l adds a computational cost of $C(g_n)$ or $C(g_l)$, respectively. With $\mathcal{O}(n)$ such evaluations per agent, this adds $\mathcal{O}(n \cdot C(g))$ operations per agent and $\mathcal{O}(n^2 \cdot C(g))$ across all agents, where $C(g) = \max\{C(g_n), C(g_l)\}$.

For a single iteration, the complexity is dominated by the state update step, resulting in $\mathcal{O}(n^2)$ when g_n and g_l are linear, and $\mathcal{O}(n^2 + n^2 \cdot C(g)) = \mathcal{O}(n^2 \cdot (1 + C(g)))$ when they are nonlinear.

Over k iterations, the total computational complexity becomes $\mathcal{O}(kn^2)$ for the linear case and $\mathcal{O}(kn^2 \cdot (1 + C(g)))$ for the nonlinear case, which can be simplified to $\mathcal{O}(kn^2 + kn^2 \cdot C(g))$ or $\mathcal{O}(kn^2 + kn \cdot C(g))$ depending on whether we consider the per-agent or network-wide complexity for the nonlinear function evaluations. $\square$

Remark 4 (Resilience to Network Disruptions). *Algorithm 1 significantly advances the state-of-the-art in distributed resource allocation through two critical innovations that enhance resilience to network disruptions:*

1. **Relaxed Weight Matrix Requirements:** *Our proposed dynamics in (15) require only that the weight matrix W be symmetric, eliminating the more restrictive weight-stochastic condition imposed by previous approaches [13, 12, 48, 30, 29]. This relaxation provides substantial benefits during link failures:*

- *No weight compensation strategies are needed when links fail or network topology changes;*
- *Unlike stochastic weight matrices that necessitate complex redistribution mechanisms [49, 50] after topology changes; and*
- *Weight-balanced conditions are naturally preserved under link removals, maintaining algorithm stability.*

Figure 1 illustrates this advantage: when the red link fails, the network loses stochasticity (observe weights at nodes v_1 and v_2), but maintains weight-balance, allowing our algorithm to proceed without reconfiguration.

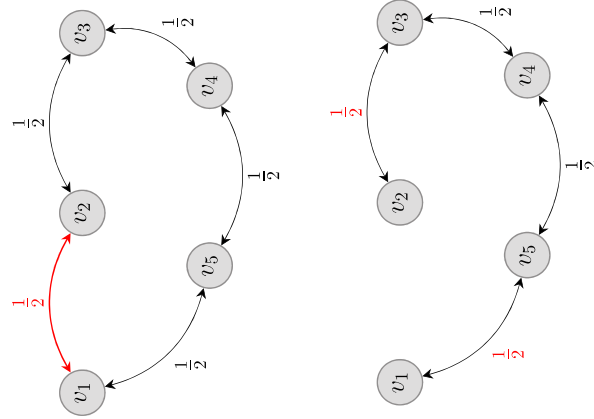


Figure 1: Comparison of network properties under link failure. Left: Original network with both weight-balance and weight-stochasticity properties. Right: After failure of the red link, weight-stochasticity is compromised at nodes v_1 and v_2 , while weight-balance is maintained, illustrating our algorithm’s inherent resilience to topology changes.

2. **Uniform Connectivity Guarantee:** *Our algorithm converges under the significantly weaker condition of uniform connectivity rather than requiring all-time connectivity as in [13, 10, 48, 26, 27, 28, 29, 30, 12]. This advancement enables the following:*

- *Tolerance of temporary network disconnections caused by high packet drop rates;*
- *Successful operation in sparse or intermittently connected network topologies; and*
- *Guaranteed convergence even when connectivity is only established periodically every $T \in \mathbb{Z}_{\geq 0}$ timesteps.*

In Section 5, we rigorously prove that the proposed relaxed connectivity requirement is sufficient to ensure algorithm convergence while dramatically improving resilience to practical network imperfections. Thus, maintaining optimal performance in realistic networking environments where disruptions are common, representing a significant practical advancement over existing approaches.

5. Convergence Analysis

In this section, we present a comprehensive convergence analysis of our proposed resilient distributed resource allocation algorithm (Algorithm 1). First, in Section 5.1, we establish fundamental convergence guarantees over uniformly-connected networks, deriving explicit bounds on the step-rate parameter to ensure optimal convergence. Next, in Section 5.2, we extend our analysis to networks experiencing random link failures, leveraging percolation theory to quantify resilience thresholds and convergence properties under various failure rates. Finally, in Section 5.3, we address the critical challenge of heterogeneous time delays across communication links, providing theoretical guarantees for convergence despite communication latency and developing practical bounds on maximum tolerable delays.

5.1. General Proof of Convergence

In the following theorem, we prove that Algorithm 1 converges to the optimal solution when the step-rate η is sufficiently small, with the specific bound on η dependent on the spectral properties λ_{2T} and λ_{nT} of the Laplacian matrix associated with the uniformly-connected network $\mathcal{G}_T$.

Theorem 1. *Under Assumptions 1 (sector-bound nonlinearity) and Assumption 2 (uniformly-connected network), Algorithm 1 converges to the optimal solution of problem (2) when the step-rate satisfies*

$$\eta < \bar{\eta} := \frac{\kappa_n \kappa_l \lambda_{2T}}{u \lambda_{nT}^2 K_n^2 K_l^2 (T+1)}, \quad (21)$$

where λ_{2T} and λ_{nT} are the smallest non-zero and largest eigenvalues of the Laplacian matrix associated with $\mathcal{G}_T$, respectively.

Proof. The all-time feasibility of the solution has been established in Lemma 3. To prove convergence, we define the residual function $\bar{F}(k) := F(\mathbf{x}(k)) - F(\mathbf{x}^*)$ and the state difference $\delta \mathbf{x}(k) := \mathbf{x}(k+T) - \mathbf{x}(k)$.

Using the smoothness property $0 < \frac{d^2 f_i}{dx_i^2} < 2u$, we can bound the change in the objective function as

$$F(\mathbf{x}(k+T)) \leq F(\mathbf{x}(k)) + \nabla F(\mathbf{x}(k))^\top \delta \mathbf{x}(k) + u \delta \mathbf{x}(k)^\top \delta \mathbf{x}(k). \quad (22)$$

To establish that the residual is non-increasing, i.e., $\bar{F}(k+T) \leq \bar{F}(k)$, it suffices to show that

$$\nabla F(\mathbf{x}(k))^\top \delta \mathbf{x}(k) + u \delta \mathbf{x}(k)^\top \delta \mathbf{x}(k) \leq 0. \quad (23)$$

For notational simplicity, we omit the time index k in the remainder of the proof. Define the dispersion parameter $\xi := \nabla F - \frac{\mathbf{1}_n^\top \nabla F}{n} \mathbf{1}_n \in \mathbb{R}^n$, which measures the deviation of the gradient from consensus.

From the proposed dynamics (15), the Laplacian-tracking property (16), and the sector-bound conditions on $g_n(\cdot)$ and $g_l(\cdot)$, we obtain

$$-\eta(T+1)\kappa_n\kappa_l L \nabla F \geq \delta \mathbf{x} \geq -\eta(T+1)K_n K_l L \nabla F. \quad (24)$$

This allows us to bound the quadratic term in (23) as follows:

$$\begin{aligned} \delta \mathbf{x}^\top \delta \mathbf{x} &\leq \eta^2(T+1)^2 K_n^2 K_l^2 \nabla F^\top L^\top L \nabla F \\ &\leq \eta^2(T+1)^2 K_n^2 K_l^2 \lambda_{nT}^2 \xi^\top \xi, \end{aligned} \quad (25)$$

where the second inequality follows from Lemma 1 and Corollary 2.

Similarly, for the first term in (23), Lemma 2 and Corollary 2 yield

$$\nabla F^\top \delta \mathbf{x} \leq -\kappa_n \kappa_l \eta(T+1) \lambda_{2T} \xi^\top \xi. \quad (26)$$

Combining (25) and (26), the condition in (23) is satisfied when

$$(-\kappa_n \kappa_l \eta \lambda_{2T}(T+1) + u \lambda_{nT}^2 K_n^2 K_l^2 \eta^2(T+1)^2) \xi^\top \xi \leq 0. \quad (27)$$

This inequality holds strictly when

$$\eta < \frac{\kappa_n \kappa_l \lambda_{2T}}{u \lambda_{nT}^2 K_n^2 K_l^2 (T+1)}. \quad (28)$$

Thus, under the stated condition on the step-rate η , the residual $\bar{F}(k)$ serves as a discrete Lyapunov function that decreases monotonically. The equality in (28) is attained only when $\xi = \mathbf{0}_n$, which corresponds to the equilibrium condition $\nabla F \in \text{span}(\mathbf{1}_n)$; thus, completing the proof. $\square$

Note that all the parameters involved in Eq. (28), including the ones related to the nonlinear mappings, are readily obtainable and thus one can easily select the step size η for convergence.

Remark 5. *The literature on distributed systems and consensus algorithms provides various characterizations of Laplacian spectral properties based on graph topology [34, 51]. For instance, [34] relates the spectral range to node degrees, while for a network with diameter $e_{\mathcal{G}}$, a lower bound on algebraic connectivity can be established as $\lambda_2(\mathcal{G}) \geq \frac{1}{n e_{\mathcal{G}}}$ [51, p. 571]. These spectral bounds are particularly valuable for predicting algorithm performance in practical network deployments.*

Notice that the convergence rate of Algorithm 1 is governed by three key factors: (i) the step-rate parameter η , which must be properly calibrated according to Theorem 1; (ii) the network connectivity quantified by the algebraic connectivity λ_2 , with denser networks exhibiting faster convergence; and (iii) the sector-bound parameters of the nonlinear functions g_n and g_l , which affect information flow through the network. These relationships provide practical guidance for tuning algorithm parameters to achieve desired performance in specific application scenarios. For the general case, the convergence rate of the proposed dynamics depends on the properties of the nonlinear mappings g_n, g_l and heterogeneous time-delays. Therefore, the exact convergence rate cannot be easily obtained.

5.2. Convergence under Link Failure

In this section, we consider the general case of random link failure in network topology, which frequently occurs due to packet drops in practical communication systems.

Similar to the modelling of time-delays discussed in Section 4.2, we model packet drops in network $\mathcal{G}(k)$ at time k as changes in the network topology; specifically, as temporary link removals from the topology. These links may disappear and reconnect dynamically, with different links potentially failing at different time instants.

The key insight in our approach is that uniform network connectivity can be maintained despite link failures if the remaining network preserves connectivity over T iterations, ensuring that $\mathcal{G}_T(k)$ remains connected. This assumption on uniform-connectivity of $\mathcal{G}_T(k)$ is central to our analysis of resilience against link failures.

Recall that the union network $\mathcal{G}_T(k) = \cup_k^{k+T} \mathcal{G}(k)$ represents the combined topology of all links present at any point over the time interval T . For a given T , a link between nodes i and j is removed from $\mathcal{G}_T(k)$ only if it is absent in all $T + 1$ network instances $\mathcal{G}(k), \mathcal{G}(k + 1), \dots, \mathcal{G}(k + T)$. In the extreme case, one can conceptualize the union network $\mathcal{G}_T(k)$ as having $T + 1$ potential instances of each link between nodes i and j . A link (i, j) is removed from $\mathcal{G}_T(k)$ only if all these $T + 1$ instances fail.

Now, let p_l denote the probability of link failure in $\mathcal{G}(k)$ at any given time. Intuitively, the probability that link (i, j) fails in the union network $\mathcal{G}_T(k)$ is p_l^{T+1} , representing the likelihood of failure across all $T + 1$ time instances.

Following the definition of bond-percolation threshold p_c from Section 3.2, when $p_l < p_c$, the network $\mathcal{G}(k)$ remains connected with probability 1 despite

random link failures. However, the challenging case occurs when $p_l > p_c$, where instantaneous connectivity is lost. This scenario is addressed in the following theorem.

Theorem 2 (Resilience to Link Failures). *Let $\mathcal{G}(k)$ be a connected network with bond-percolation threshold p_c . Under random link failures occurring with probability p_l , where $p_c < p_l < 1$, the union network $\mathcal{G}_T(k)$ maintains uniform connectivity with probability 1 for any time window T satisfying*

$$p_l^{T+1} < p_c. \quad (29)$$

Proof. Given a network with bond-percolation threshold p_c , we first establish the existence of a minimum time window T^* that ensures continued connectivity despite link failures occurring with probability $p_l > p_c$.

Let $T^* \in \mathbb{Z}_{>0}$ denote the minimum value satisfying $p_l^{1+T^*} < p_c$. Such a value must exist since (i) $p_l < 1$ by assumption, making p_l^{1+T} strictly less than 1 for all $T > 0$, (ii) $\lim_{T \rightarrow \infty} p_l^{1+T} = 0$ for $p_l < 1$, and (iii) $p_c > 0$ by the definition of percolation threshold.

The function p_l^{1+T} has two important properties: (a) it is monotonically decreasing with respect to T (since $p_l < 1$), and (b) it is monotonically increasing with respect to p_l (for any fixed $T > 0$).

Therefore, for any $T \geq T^*$, we have $p_l^{1+T} \leq p_l^{1+T^*} < p_c$. This implies that the probability of a link failing in all $T+1$ consecutive time steps is less than the percolation threshold p_c .

By the definition of bond percolation (Section 3.2), when the link failure probability is less than p_c , the network remains connected with probability 1. Consequently, for any $T \geq T^*$ and $p_c < p_l < 1$, the union network $\mathcal{G}_T(k)$ maintains connectivity with probability 1, thereby ensuring uniform connectivity over the time window T . $\square$

Corollary 3 (Convergence Under Link Failure). *For a network experiencing random link failures with probability p_l where $p_c < p_l < 1$, Algorithm 1 converges to the optimal solution with probability 1 when*

1. *The time window T satisfies $p_l^{T+1} < p_c$ as per Theorem 2, and*
2. *The step-rate η satisfies the bound in Equation (28) from Theorem 1.*

The resilience of our algorithm to link failures comes with a performance trade-off. As the link failure probability p_l increases, the network becomes

more sparse, reducing the algebraic connectivity λ_2 . This spectral degradation has direct implications on algorithm performance; specifically, (28) indicates that smaller values of λ_2 necessitate smaller step-rates η to maintain convergence guarantees, resulting in slower overall convergence.

Hence, the relationship between link failure rate and convergence speed establishes a quantifiable resilience-performance trade-off that can guide parameter selection in practical deployments. The algorithm can tolerate substantial link failures while maintaining convergence guarantees, albeit at reduced convergence rates that are precisely characterized by the spectral properties of the resulting network topology.

5.3. Convergence under Time-Delays

Latency is an unavoidable challenge in practical communication networks, arising from packet retransmission, processing bottlenecks, and physical transmission constraints. This section analyzes the convergence properties of our proposed algorithm under heterogeneous time delays across the multi-agent network.

For the dynamics defined in (19), we can extend our theoretical analysis by examining the union network $\mathcal{G}_{T+\bar{\tau}}(k) = \cup_{j=k}^{k+T+\bar{\tau}} \mathcal{G}(j)$. This formulation reveals a crucial insight: when the delay-free network $\mathcal{G}_T(k)$ exhibits connectivity over T time steps, we can establish connectivity of $\mathcal{G}_{T+\bar{\tau}}(k)$ even when communication is subject to heterogeneous time delays bounded by $\bar{\tau}$. This perspective allows us to address arbitrary, time-varying, and heterogeneous delays within a unified theoretical framework.

Theorem 3. *Given a uniformly-connected network $\mathcal{G}_T(k)$, Algorithm 1 converges to the optimal solution of problem (2) for a sufficiently small step-rate η_τ satisfying*

$$\eta_\tau < \frac{\kappa_n \kappa_l \lambda_{2T}}{u \lambda_{nT}^2 K_n^2 K_l^2 (T + \bar{\tau} + 1)}, \quad (30)$$

where $\bar{\tau}$ represents the upper bound on heterogeneous time-delays across all communication links.

Proof. The proof extends the analysis of Theorem 1 to the uniformly-connected network $\mathcal{G}_{T+\bar{\tau}}(k)$. A key observation is that the eigenspectrum of $\mathcal{G}_{T+\bar{\tau}}(k)$ is identical to that of $\mathcal{G}_T(k)$, since the same set of communication links is distributed over an extended time scale $T + \bar{\tau}$.

Specifically, for a network subject to time delays bounded by $\bar{\tau}$, the set of links $V_{T+\bar{\tau}} = \cup_{j=k}^{k+T+\bar{\tau}} V(j)$ corresponds exactly to the set of links V_T in the

delay-free case over time period T . This topological equivalence allows us to apply the same Lyapunov analysis as in Theorem 1, with the adjustment that we consider the extended time horizon $T + \bar{\tau}$ instead of T .

Following the same sequence of algebraic manipulations with the modified time horizon, we arrive at the convergence condition in (30), which guarantees that the residual function serves as a discrete Lyapunov function that decreases monotonically. $\square$

The bound in Theorem 3 characterizes the fundamental trade-off between delay tolerance and convergence rate. For a given step-rate η_τ , we can determine the maximum tolerable delay $\bar{\tau}$ for guaranteed convergence given by

$$\bar{\tau} < \frac{\kappa_n \kappa_l \lambda_{2T}}{u \eta_\tau \lambda_{nT}^2 K_n^2 K_l^2} - 1 - T. \quad (31)$$

This expression provides the upper bound on network delays for which Algorithm 1 maintains convergence guarantees despite heterogeneous, time-varying delays. In scenarios with unknown but bounded delays, agents can synchronize their information updates at intervals of $\bar{\tau}$ time steps. Under this implementation, all communication links in $\mathcal{G}(k)$ effectively transfer information to time $k + \bar{\tau}$, preserving uniform connectivity over the original time period T . This approach enables a tighter upper bound on the step-rate, analogous to the bound in (28).

Remark 6. Equation (31) reveals a fundamental resilience versus performance trade-off. Specifically, accommodating larger time delays $\bar{\tau}$ necessitates reducing the step-rate η_τ to maintain convergence guarantees. This reduction in step-rate inevitably leads to slower convergence. This trade-off provides system designers with a quantitative framework for balancing delay tolerance against convergence speed in practical deployments.

The analysis presented in this subsection demonstrates that our algorithm can tolerate substantial communication delays while maintaining both feasibility and optimality guarantees – a critical advancement over existing approaches that require either continuous connectivity or synchronous, delay-free communication.

6. Simulation

In this section, we provide simulations to validate the performance of Algorithm 1 in the presence of link failure and time delays. For the simulation

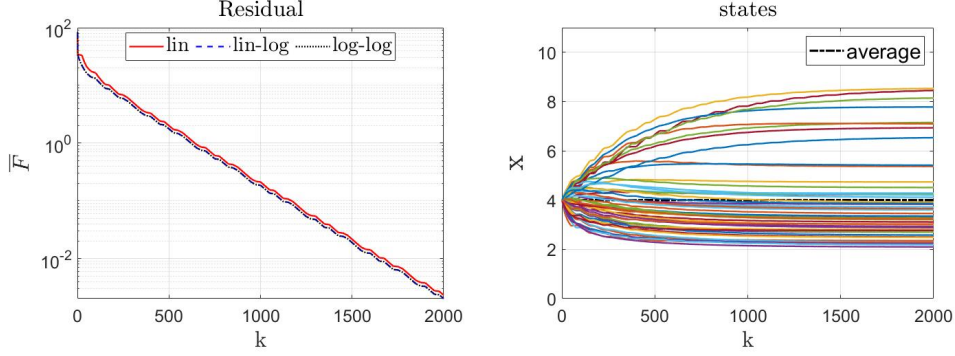


Figure 2: Convergence behaviour under dynamic network topology. Left: Evolution of cost residual with penalty term $\sigma([x_i - M_i]^+ + [m_i - x_i]^+)$ for both linear and nonlinear (log-scale quantization) models in a non-persistently connected network. Right: State trajectories of all agents under log-scale quantization, demonstrating constant average values and confirming the all-time feasibility of the proposed solution.

we optimize, in a distributed way, the following cost function (taken from [15]) at every agent i :

$$f_i(x_i) = \omega_i(x_i - \alpha_i)^4 + \sigma([x_i - M_i]^+ + [m_i - x_i]^+), \quad (32)$$

with penalty terms $\sigma([x_i - M_i]^+ + [m_i - x_i]^+)$ and $[u]^+ = (\max\{0, u\})^2$ addressing the box constraints. For the simulation, we consider $\sigma = 20$ large enough to better address the box constraints with $m_i = 1$, $M_i = 10$. Other simulation parameters are chosen as follows: $\alpha_i \in (0, 2]$, $\omega_i \in (0, 0.02]$, and $b = 200$ as the sum of resources.

6.1. Convergence under Dynamic Network Topology

The multi-agent network is considered as an ER random graph of $n = 50$ nodes with four different linking probability $p \in \{20\%, 10\%, 5\%, 1\%\}$ to model a dynamic network with random entries W_{ij} . Note that, for $p = 1\%$, we have $p < \frac{1}{n-1}$ implying that the network is disconnected, since the network switches between these four possible topologies every 25 iterations.

For the sector-bound nonlinearity, we consider log-scale quantization defined as $q(x) = \text{sgn}(x) \exp(\epsilon \lceil \frac{\log(|x|)}{\epsilon} \rceil)$ with ϵ as the quantization level. It is known that logarithmic quantization satisfies $x(1 - \frac{\epsilon}{2}) \leq q(x) \leq x(1 + \frac{\epsilon}{2})$.

The distributed solution follows (15) with three different options of nonlinearity at the node and link dynamics: (i) linear-linear: $g_n(x) = x$ and

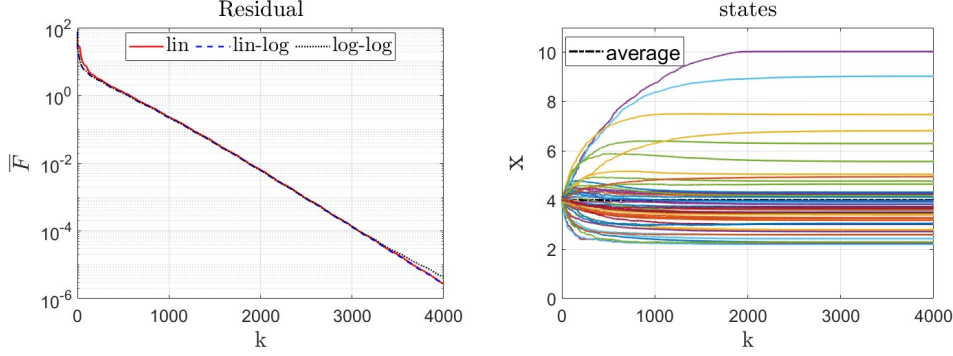


Figure 3: Algorithm performance with logarithmic penalty function. Left: Convergence of cost residual with smoothed penalty term $\frac{1}{\mu}(\log(1 + \exp(\mu(x_i - M_i))) + \log(1 + \exp(\mu(m_i - x_i))))$ under dynamic network conditions. Right: State evolution of all agents maintaining constant average value, confirming all-time feasibility with logarithmic constraints.

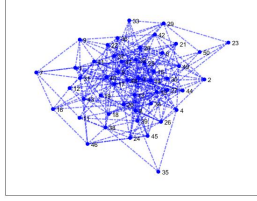
$g_l(x) = x$; (ii) linear-logarithmic: $g_n(x) = x$ and $g_l(x) = q(x)$; and (iii) logarithmic-logarithmic: $g_n(x) = q(x)$ and $g_l(x) = q(x)$.

For our implementation, we choose $\epsilon_l = \frac{1}{8}$ and $\epsilon_n = \frac{1}{1024}$ as the link and node quantization levels, respectively. We set $\eta = 0.1$ for the simulation, and the results are shown in Fig. 2. The average of states $\frac{1}{n} \sum_{i=1}^n x_i$ remains constant equal to $\frac{b}{n}$ throughout the evolution of dynamics, demonstrating the all-time feasibility of the proposed solution, i.e., $\sum_{i=1}^n x_i(t) = b$ for $t \geq 0$.

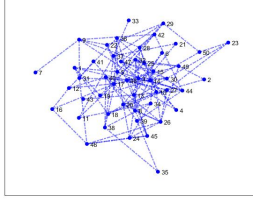
We provide an additional simulation using a logarithmic penalty term in the form $\frac{1}{\mu}(\log(1 + \exp(\mu(x_i - M_i))) + \log(1 + \exp(\mu(m_i - x_i))))$ to address the box constraints. For this penalty term, we set $\mu = 5$, which is sufficiently large to properly enforce the box constraints. All other simulation parameters remain the same as in the previous case. The results are presented in Fig. 3.

6.2. Resilience to Link Failure

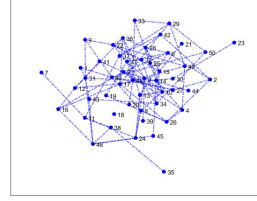
Now, we examine an Erdős–Rényi (ER) network with linking probability $p = 20\%$ subjected to various link failure rates (i.e., $p_l \in \{50\%, 70\%, 85\%, 92\%\}$) to systematically evaluate network resilience. The bond-percolation threshold for this graph configuration is $p_c = 79.5\%$, as determined by (14). This threshold creates two distinct operational regimes: for $p_l < p_c$, network connectivity is preserved with probability approaching 1 despite random link failures; for $p_l > p_c$, the network loses instantaneous connectivity but maintains uniform connectivity over extended time windows T that satisfy the conditions in Theorem 2 and (29).



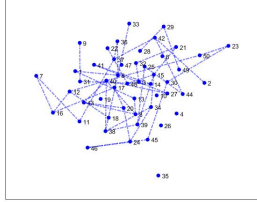
(a) $p_l = 0$



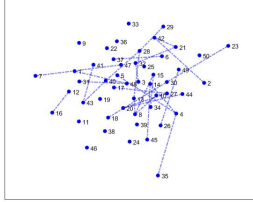
(b) $p_l = 0.5$



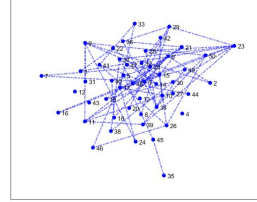
(c) $p_l = 0.7$



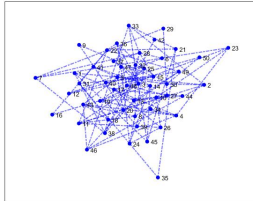
(d) $p_l = 0.85$



(e) $p_l = 0.92$



(f) $p_l = 0.85, T = 2$



(g) $p_l = 0.92, T = 4$

Figure 4: Network topology under varying link failure conditions. (a) Original Erdős–Rényi network with 20% linking probability. (b)-(c) Networks with failure rates below percolation threshold ($p_l < p_c$), maintaining connectivity with probability 1. (d)-(e) Networks with failure rates exceeding percolation threshold ($p_l > p_c$), becoming disconnected. (f)-(g) Union networks over $T = 2$ and $T = 4$ time steps respectively, demonstrating uniform connectivity when $p_l^{T+1} < p_c$ despite instantaneous disconnections.

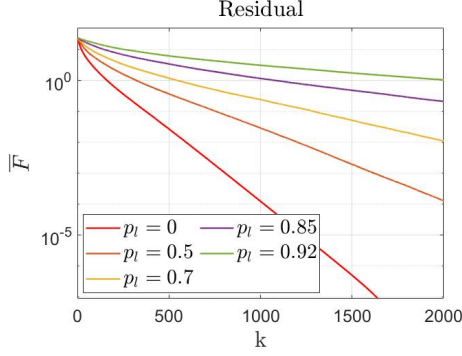


Figure 5: Impact of link failure probability on convergence performance. The plot shows cost residual evolution under varying link failure rates (50%-92%), demonstrating that higher failure probabilities require longer time windows for uniform connectivity, resulting in slower but still guaranteed convergence to optimal solutions.

Specifically, when $p_l = 85\%$ and $p_l = 92\%$, uniform connectivity is maintained over $T = 2$ and $T = 4$ time steps, respectively. Figure 4 illustrates representative network topologies under these different failure probabilities, demonstrating the structural changes that occur as failure rates increase.

To simulate realistic packet drops, our model implements dynamic link failures at each time step. Links randomly disconnect (simulating packet drops) and later reconnect (representing successful packet delivery), while other links may simultaneously fail with probability p_l . This creates a highly dynamic network topology with continuously changing connections between agents. As expected, higher link failure rates result in slower convergence of our resilient dynamics due to reduced information exchange across the network, as demonstrated in Figure 5 with step rate $\eta = 0.2$.

6.3. Resilience to Time-Delay

Next, we evaluate the resilience of our distributed resource allocation algorithm to communication delays by implementing the time-delayed dynamics in Equation (19). Our testbed uses the same ER network configuration (with linking probability $p = 20\%$) while introducing heterogeneous time-varying delays across links. We conduct experiments with maximum delay bounds $\bar{\tau} \in \{2, 4, 6\}$ time steps to analyze performance degradation under increasing latency conditions.

For this network topology, spectral analysis yields $\lambda_2 = 0.044$ and $\lambda_n = 0.311$, with sector-bound parameters $\kappa_n = 1$, $\kappa_l = 0.938$, $\mathcal{K}_n = 1$, $\mathcal{K}_l = 1.062$,

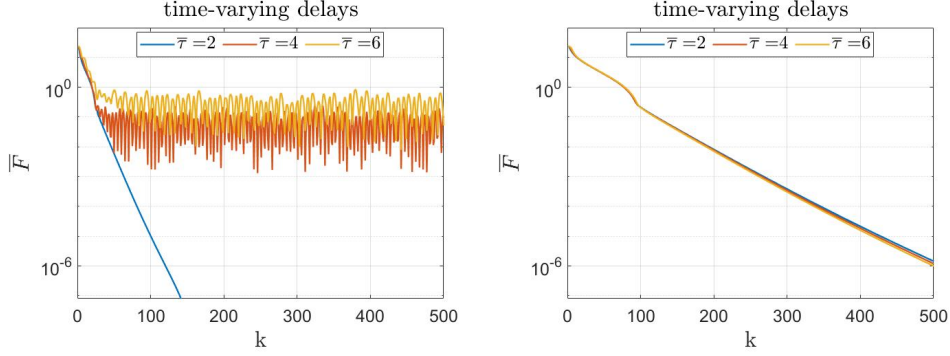


Figure 6: Algorithm resilience to communication delays. Left: Convergence behaviour with step-rate $\eta = 2$, showing successful convergence for delays $\bar{\tau} = 2$ but divergence for larger delays. Right: Improved resilience with reduced step-rate $\eta = 0.5$, achieving convergence for all tested delay bounds ($\bar{\tau} \in \{2, 4, 6\}$), illustrating the fundamental trade-off between delay tolerance and convergence speed.

and smoothness parameter $u \approx 0.115$. Substituting these values into the theoretical bound provided by (31), we determine that with step-rate $\eta = 2$, convergence is guaranteed only for $\bar{\tau} \leq 2$.

This theoretical prediction is validated by our experimental results shown in Figure 6(left), where convergence is observed for $\bar{\tau} = 2$ but not for $\bar{\tau} = 4$ or $\bar{\tau} = 6$. Following the relationship established in (31) and discussed in Remark 6, we can accommodate larger time delays by reducing the step-rate. When decreasing η to 0.5, as illustrated in Figure 6(right), the algorithm successfully converges for all tested delay bounds $\bar{\tau} \in \{2, 4, 6\}$, though at a correspondingly slower convergence rate – demonstrating the fundamental trade-off between delay tolerance and convergence speed.

6.4. Practical Engineering Example and Comparison with the Literature

As a practical engineering application, we demonstrate our algorithm on the distributed economic dispatch problem in power systems, where generation resources must be optimally allocated across a network of interconnected generators. This problem, extensively studied in [10, 52], requires distributing a total power demand of $b = 600$ MW among $n = 10$ generators while minimizing generation costs.

Each generator i has a quadratic cost function that models its operational expenses described by

$$f_i(p_i) = a_i p_i^2 + b_i p_i + c_i, \quad (33)$$

where p_i represents the power output (in MW) of generator i , and coefficients a_i , b_i , and c_i characterize the specific cost profile of each generator. These parameters are configured according to realistic generator data provided in [52].

Practical operating constraints require that each generator’s output remains within its capacity limits, represented as $m_i \leq p_i \leq M_i$, where $m_i = 20$ MW and $M_i = 110$ MW denote the minimum and maximum generation capacities, respectively. We enforce these constraints through penalty terms of the form $\sigma([p_i - M_i]^+ + [m_i - p_i]^+)$, where $[u]^+ = (\max\{0, u\})^2$ and $\sigma = 40$ is a sufficiently large penalty coefficient.

We implement Algorithm 1 with both log-scale quantization for data transmission and sign-based nonlinearity to enhance convergence performance. Figure 7 illustrates the evolution of both the cost residual and the power allocations over time. For comparative analysis, we benchmark our method’s convergence rate against several state-of-the-art approaches: standard linear methods [10, 52], acceleration-based techniques [53], finite-time convergent algorithms [54], and fixed-time convergent solutions [55]. For all these allocation techniques, the network topology is considered as an ER network with linking probability of $p = 20\%$. As a comparison, although the approach proposed by [53] reaches faster convergence using momentum-based linear dynamics, it cannot address nonlinear mappings (e.g., due to log-scale quantization and saturation), cannot handle communication time-delays over the network, and only converges over fixed and time-invariant networks. On the other hand, our proposed approach handles nonlinear mapping and is resilient to communication time-delays and link failures by addressing convergence over uniformly-connected time-varying networks. As compared to other solutions, the convergence of our proposed solution is comparable and even better in some cases. It is worth mentioning that the convergence rate of our proposed algorithm can be improved by using nonlinear signum-based mappings at the node dynamics.

To evaluate our algorithm’s resilience to communication imperfections, we conduct extensive simulations under two critical network disruption scenarios: (i) stochastic link failures and (ii) heterogeneous communication delays.

For the link failure analysis, we examine the algorithm’s performance under progressively deteriorating network conditions with failure probabilities $p_l \in \{50\%, 70\%, 85\%, 92\%\}$. This range strategically spans both sub-critical ($p_l < p_c$) and super-critical ($p_l > p_c$) regimes relative to the network’s percolation threshold p_c .

Similarly, for communication delay analysis, we test our algorithm’s ro-

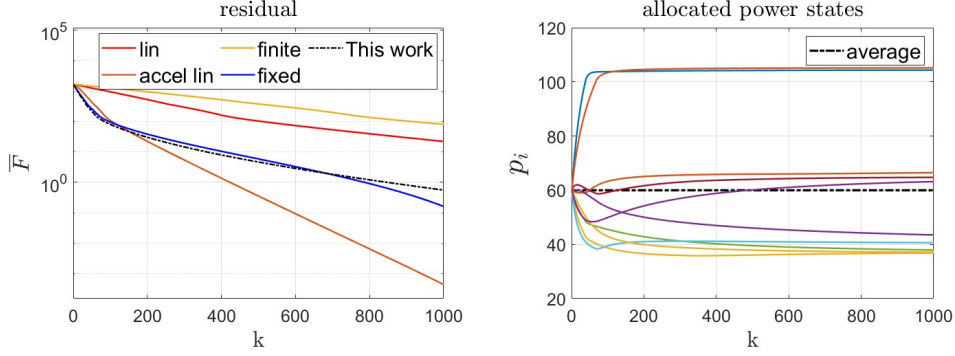


Figure 7: Economic dispatch performance comparison. Left: Convergence rates of our proposed algorithm versus existing approaches (linear, accelerated linear, finite-time, and fixed-time methods) for power allocation optimization. Right: Evolution of individual generator outputs, maintaining constant sum throughout the iteration process, demonstrating strict adherence to power demand constraints at all times.

bustness under heterogeneous time delays bounded by maximum values $\bar{\tau} \in \{3, 7, 10\}$ time steps. These delays are implemented as random, time-varying quantities that differ across communication links, accurately modelling real-world network conditions.

Figure 8 presents the convergence behaviour under these challenging scenarios, demonstrating that our algorithm maintains convergence despite significant communication impairments. This robust performance stands in stark contrast to existing approaches in the literature [10, 52, 53, 54, 55], none of which provide theoretical guarantees for convergence under both time-delayed communications and uniformly-connected network topologies.

The comparative advantage of our approach is particularly evident in practical deployment scenarios where packet drops and communication delays are inevitable. While conventional algorithms fail to maintain convergence in such environments, our solution demonstrates reliable performance across varying degrees of link failures and communication delays, confirming both the theoretical resilience guarantees established in Theorems 2 and 3 and the practical utility of our approach for real-world distributed systems.

7. Conclusions

This paper introduces a novel framework for resilient distributed resource allocation that fundamentally advances the state-of-the-art in networked

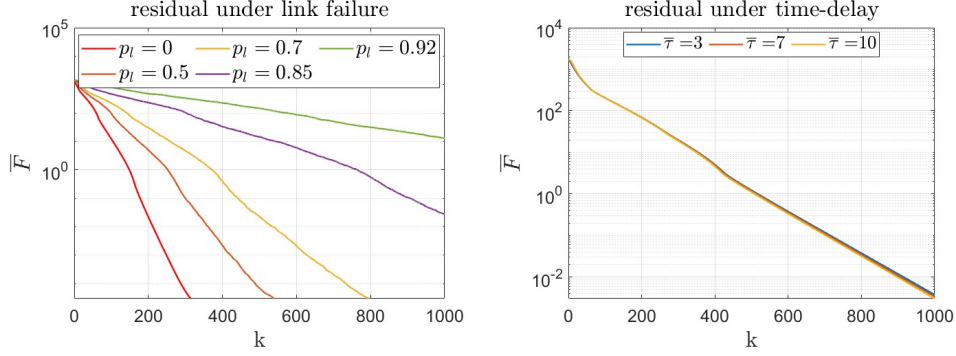


Figure 8: Resilience of economic dispatch under communication imperfections. Left: Power allocation convergence under various link failure rates, demonstrating robust performance even with severe network disruptions. Right: Convergence behaviour with heterogeneous time delays, showing the algorithm’s ability to maintain optimality despite communication latency in practical power systems.

multi-agent systems operating under challenging communication conditions. Our approach makes three significant theoretical and practical contributions:

First, we have developed a distributed algorithm that maintains all-time feasibility for resource-demand balance – a critical property ensuring system stability even during network disruptions or early termination. Unlike existing approaches that provide only asymptotic feasibility, our solution guarantees constraint satisfaction at every iteration.

Second, our algorithm achieves provable convergence to optimal solutions despite random link failures and heterogeneous communication delays. By leveraging percolation theory and spectral graph analysis, we established precise mathematical bounds on tolerable failure rates and delay thresholds, providing system designers with concrete guidelines for implementation in real-world networks.

Third, extensive simulations across various network configurations – including dynamic topologies, severe link failures (up to 92%), and substantial communication delays – demonstrate the algorithm’s remarkable resilience in practical scenarios such as economic dispatch in power systems.

One direction of future research is to extend the results to consider DRA over directed network topologies. The theoretical framework developed in this paper extends beyond resource allocation, providing a foundation for enhancing resilience in various distributed systems. Future research can build upon our approach to address emerging challenges in distributed machine

learning (particularly federated learning under unreliable communication), multi-agent reinforcement learning, and edge computing applications where network reliability cannot be guaranteed. Additionally, our methodology could be adapted to handle adversarial attacks and strategic network disruptions, further advancing the robustness of distributed systems in security-critical applications.

Acknowledgement

This work is funded by Semnan University, grant No. 226/1403/1403214.

References

- [1] M. Pirani, A. Mitra, S. Sundaram, Graph-theoretic approaches for analyzing the resilience of distributed control systems: A tutorial and survey, *Automatica* 157 (2023) 111264.
- [2] M. Doostmohammadian, A. Aghasi, M. Pirani, E. Nekouei, H. Zarrabi, R. Keypour, A. I. Rikos, K. H. Johansson, Survey of distributed algorithms for resource allocation over multi-agent systems, *Annual Reviews in Control* 59 (2025) 100983.
- [3] M. Doostmohammadian, A. Aghasi, A. I. Rikos, A. Grammenos, E. Kalyvianaki, C. N. Hadjicostis, K. H. Johansson, T. Charalambous, Distributed CPU Scheduling Subject to Nonlinear Constraints, in: *IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 746–751, 2022.
- [4] G. Oliva, T. Charalambous, L. Faramondi, R. Setola, A. Gasparri, Best Effort Workload Disparity Minimization in Multi-Agent Systems with Capacity Constraints, *IEEE Transactions on Automatic Control* 68 (8) (2022) 4998–5005.
- [5] M. Doostmohammadian, H. Sayyaadi, M. Moarref, A novel consensus protocol using facility location algorithms, in: *IEEE Conf. on Control Applications & Intelligent Control*, 914–919, 2009.
- [6] B. Telsang, Decentralized Resource Allocation through Constrained Centroidal Voronoi Tessellations, Ph.D. thesis, University of Tennessee, 2022.

- [7] Q. Kang, H. He, H. Song, R. Deng, Task allocation for maximizing reliability of distributed computing systems using honeybee mating optimization, *Journal of Systems and Software* 83 (11) (2010) 2165–2174.
- [8] P. Yin, S. Yu, P. Wang, Y. Wang, Task allocation for maximizing reliability of a distributed system using hybrid particle swarm optimization, *Journal of Systems and Software* 80 (5) (2007) 724–735.
- [9] X. Zhou, Z. Ma, S. Zou, J. Zhang, Y. Guo, Distributed Energy Management of Double-Side Multienergy Systems via Sub-Gradient Averaging Consensus, *IEEE Transactions on Smart Grid* 14 (2) (2022) 979–995.
- [10] A. Cherukuri, J. Cortés, Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment, *Automatica* 74 (2016) 183–193.
- [11] S. Shah, G. P. Incremona, P. Bolzern, P. Colaneri, Optimization based AIMD saturated algorithms for public charging of electric vehicles, *European Journal of Control* 47 (2019) 74–83.
- [12] A. Falsone, I. Notarnicola, G. Notarstefano, M. Prandini, Tracking-ADMM for distributed constraint-coupled optimization, *Automatica* 117 (2020) 108962.
- [13] L. Xiao, S. Boyd, Optimal scaling of a gradient method for distributed resource allocation, *Journal of Optimization Theory and Applications* 129 (3) (2006) 469–488.
- [14] J. Zhang, K. You, K. Cai, Distributed dual gradient tracking for resource allocation in unbalanced networks, *IEEE Transactions on Signal Processing* 68 (2020) 2186–2198.
- [15] T. T. Doan, C. L. Beck, Distributed Lagrangian methods for network resource allocation, in: *IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 650–655, 2017.
- [16] A. Rogozin, D. Yarmoshik, K. Kopylova, A. Gasnikov, Decentralized Strongly-Convex Optimization with Affine Constraints: Primal and Dual Approaches, in: *International Conference on Optimization and Applications*, Springer, 93–105, 2022.

- [17] W. Lin, Y. Wang, C. Li, X. Yu, Predefined-time optimization for distributed resource allocation, *J. of the Franklin Institute* 357 (16) (2020) 11323–11348.
- [18] X. Wang, C. Su, H. Dai, L. Yan, Predefined-time distributed optimization algorithms for a class of resource allocation problem, *Journal of the Franklin Institute* 361 (12) (2024) 107009.
- [19] G. Shao, R. Wang, X. Wang, K. Liu, Distributed algorithm for resource allocation problems under persistent attacks, *Journal of the Franklin Institute* 357 (10) (2020) 6241–6256.
- [20] X. Cai, H. Zhong, Y. Li, J. Liao, X. Chen, X. Nan, B. Gao, An event-triggered quantization communication strategy for distributed optimal resource allocation, *Systems & Control Letters* 180 (2023) 105619.
- [21] T. Zhou, H. Wu, J. Cao, Distributed optimization in predefined-time for multi-agent systems over a directed network, *Information Sciences* 615 (2022) 743–757.
- [22] Y. Zhu, G. Wen, W. Yu, X. Yu, Continuous-Time Distributed Proximal Gradient Algorithms for Nonsmooth Resource Allocation Over General Digraphs, *IEEE Transactions on Network Science and Engineering* 8 (2) (2021) 1733–1744.
- [23] Y. Zhu, W. Ren, W. Yu, G. Wen, Distributed resource allocation over directed graphs via continuous-time algorithms, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51 (2) (2019) 1097–1106.
- [24] C. N. Hadjicostis, T. Charalambous, Average consensus in the presence of delays in directed graph topologies, *IEEE Transactions on Automatic Control* 59 (3) (2013) 763–768.
- [25] X. Wang, Y. Hong, X. Sun, K. Liu, Distributed optimization for resource allocation problems under large delays, *IEEE Transactions on Industrial Electronics* 66 (12) (2019) 9448–9457.
- [26] M. Doostmohammadian, W. Jiang, T. Charalambous, DTAC-ADMM: Delay-Tolerant Augmented Consensus ADMM-based Algorithm for Distributed Resource Allocation, in: *IEEE 61st Conference on Decision and Control (CDC)*, IEEE, 308–315, 2022.

- [27] G. Banjac, F. Rey, P. Goulart, J. Lygeros, Decentralized resource allocation via dual consensus ADMM, in: American Control Conference (ACC), IEEE, 2789–2794, 2019.
- [28] W. Jiang, M. Doostmohammadian, T. Charalambous, Distributed resource allocation via ADMM over digraphs, in: IEEE 61st Conference on Decision and Control, IEEE, 5645–5651, 2022.
- [29] Z. Wang, J. Xu, S. Zhu, C. Chen, A dual splitting method for distributed economic dispatch in multi-energy systems, IFAC-PapersOnLine 53 (2) (2020) 12566–12571.
- [30] R. Carli, M. Dotoli, Distributed Alternating Direction Method of Multipliers for Linearly Constrained Optimization Over a Network, IEEE Control Systems Letters 4 (1) (2020) 247–252, doi:\bibinfo{doi}{10.1109/LCSYS.2019.2923078}.
- [31] Y. Nesterov, Introductory lectures on convex programming, volume I: Basic course, Lecture notes 3 (4) (1998) 5.
- [32] D. Bertsekas, A. Nedic, A. Ozdaglar, Convex Analysis and Optimization, Athena Scientific, Belmont, MA, 2003.
- [33] A. Barabási, Network science, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 371 (1987) (2013) 20120375.
- [34] R. Olfati-Saber, R. M. Murray, Consensus problems in networks of agents with switching topology and time-delays, IEEE Transactions on Automatic Control 49, no. 9 (2004) 1520–1533.
- [35] R. A. Horn, C. R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 1987.
- [36] R. Cohen, K. Erez, D. Ben-Avraham, S. Havlin, Resilience of the internet to random breakdowns, Physical review letters 85 (21) (2000) 4626.
- [37] M. Li, R. Liu, L. Lü, M. Hu, S. Xu, Y. Zhang, Percolation on complex networks: Theory and application, Physics Reports 907 (2021) 1–68.

- [38] M. Sahimi, Applications of percolation theory, CRC Press, 1994.
- [39] J. Wierman, Percolation theory, Wiley Encyclopedia of Operations Research and Management Science 1.
- [40] A. Mohseni-Kabir, M. Pant, D. Towsley, S. Guha, A. Swami, Percolation thresholds for robust network connectivity, *Journal of Statistical Mechanics: Theory and Experiment* 2021 (1) (2021) 013212.
- [41] P. Erdos, A. Rényi, On the evolution of random graphs, *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* 5 (1) (1960) 17–60.
- [42] H. Kawamoto, H. Takayasu, H. J. Jensen, M. Takayasu, Precise calculation of a bond percolation transition and survival rates of nodes in a complex network, *PloS one* 10 (4) (2015) e0119979.
- [43] B. Bollobás, *Modern Graph Theory*, Springer Verlag, New York, NY, 1998.
- [44] M. Doostmohammadian, M. I. Qureshi, M. H. Khalesi, H. R. Rabiee, U. A. Khan, Log-scale quantization in distributed first-order methods: Gradient-based learning from distributed data, *IEEE Transactions on Automation Science and Engineering* 22 (2025) 10948–10959.
- [45] M. Doostmohammadian, H. R. Rabiee, Momentum-based Accelerated Algorithm for Distributed Optimization under Sector-Bound Nonlinearity, *Journal of the Franklin Institute* 362 (2025) 107857.
- [46] X. Li, K. Lin, L. Li, Y. Hong, J. Chen, On faster convergence of scaled sign gradient descent, *IEEE Transactions on Industrial Informatics* 20 (2) (2023) 1732–1741.
- [47] A. Cherukuri, J. Cortés, Distributed generator coordination for initialization and anytime optimization in economic dispatch, *IEEE Transactions on Control of Network Systems* 2 (3) (2015) 226–237.
- [48] A. I. Rikos, A. Grammenos, E. Kalyvianaki, C. N. Hadjicostis, T. Charalambous, K. H. Johansson, Optimal CPU scheduling in data centers via a finite-time distributed quantized coordination mechanism, in: *60th IEEE Conference on Decision and Control*, IEEE, 6276–6281, 2021.

- [49] N. H. Vaidya, C. N. Hadjicostis, A. D. Dominguez-Garcia, Robust average consensus over packet dropping links: Analysis via coefficients of ergodicity, in: IEEE 51st IEEE Conference on Decision and Control, 2761–2766, 2012.
- [50] F. Fagnani, S. Zampieri, Average Consensus with Packet Drop Communication, SIAM Journal on Control and Optimization 48 (1) (2009) 102–133.
- [51] J. L. Gross, J. Yellen, Handbook of Graph Theory, CRC Press, 2004.
- [52] S. Kar, G. Hug, Distributed robust economic dispatch in power systems: A consensus+ innovations approach, in: IEEE Power and Energy Society General Meeting, 1–8, 2012.
- [53] E. Ghadimi, M. Johansson, I. Shames, Accelerated gradient methods for networked optimization, in: American Control Conference, IEEE, 1668–1673, 2011.
- [54] G. Chen, J. Ren, E. N. Feng, Distributed finite-time economic dispatch of a network of energy resources, IEEE Transactions on Smart Grid 8 (2) (2016) 822–832.
- [55] B. Ning, Q. Han, Z. Zuo, Distributed optimization for multiagent systems: An edge-based fixed-time consensus approach, IEEE Transactions on Cybernetics 49 (1) (2017) 122–132.