Future Generation Computer Systems 175 (2026) 108017

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Optimization under attack: Resilience, vulnerability, and the path to collapse

Amal Aldawsari [a,b,*], Evangelos Pournaras [a]

[a] School of Computer Science, University of Leeds, Leeds, UK
[b] College of Computer Science and Engineering, University of Hail, Hail, Saudi Arabia

ARTICLE INFO

Keywords:
Optimization
Multi-agent systems
Adversary behavior
Resilience
Vulnerability
Distributed systems
Fault-tolerance

ABSTRACT

Optimization is critical for improving the operations of large-scale socio-technical infrastructures such as those found in energy, mobility, and information systems. In particular, understanding the performance of multi-agent discrete-choice combinatorial optimization under distributed adversarial attacks is a compelling and underexplored problem. Multi-agent systems involve a large number of remote control variables that can influence the cost-effectiveness of distributed optimization heuristics. This paper unravels, for the first time, the trajectories of distributed optimization from resilience to vulnerability, and finally to collapse under varying adversarial influence. Using real-world and synthetic data to generate over 112 million multi-agent optimization scenarios, we systematically assess how the number of agents with varying levels of adversarial severity and network positioning influences optimization performance, with particular attention to the impact on Pareto optimality. With this large-scale dataset, made openly available as a benchmark, we disentangle how optimization systems remain resilient to adversaries and which adversary conditions make optimization vulnerable or cause collapse. These findings can support the design of self-healing strategies for fault tolerance and fault correction, addressing a critical gap in adversarial distributed optimization.

1. Introduction

The rapid development of Internet of Things (IoT) applications has brought transformative changes in numerous domains, ranging from smart cities to industrial automation and healthcare [1]. These applications involve vast networks of interconnected devices that generate substantial amounts of data, and require efficient and distributed decision-making [2]. Distributed optimization is essential in such scenarios, as it enables multiple agents to collaborate effectively without centralized coordination. This approach ensures scalability, reliability, and robust performance across diverse applications, including energy management, autonomous systems, and federated learning [3–7].

While distributed optimization is essential for achieving system-wide objectives, most algorithms rely on assumptions of rational, cooperative, and non-adversarial agents contributing toward the global objective without prioritizing their individual goals over collective outcomes. However, real-world scenarios often deviate from these assumptions; adversary agents can disrupt optimization by introducing inaccuracies, manipulating decisions, or compromising functionality [8–11]. For example, in smart grid systems where autonomous agents (e.g., households) collaborate to manage energy distribution and prevent power outages, certain households may act adversarially by manipulating consumption data to prioritize their self-interest, i.e., their thermal comfort. Such disruptions distort energy allocation, leading to inefficient resource use and blackouts [9,10,12–14].

Several studies have examined multi-agent optimization in continuous-choice frameworks [8–11,13,15]. While stochastic optimization techniques address noise, they lack mechanisms to counteract strategic manipulations by adversarial agents [16,17]. Such manipulations amplify system vulnerability by prioritizing individual goals over collective objectives, which can lead to inefficiencies, instability, and eventual system collapse [18,19]. Despite these risks, adversarial disruptions in discrete-choice settings have received little attention.

This paper studies multi-agent systems in discrete-choice combinatorial optimization under adversarial conditions, with the aim to unravel the trajectories of resilience, vulnerability, and collapse, offering a novel framework to understand system optimization behavior. In this context, resilience refers to the system ability to maintain performance despite adversarial influence; vulnerability captures the emergence of inefficiencies due to intolerable adversarial behaviors; and collapse occurs when the system significantly under-performs as a result of failure to cope with adversarial behaviors. This study evaluates the impact of adversarial agents on optimization performance, analyzing critical parameters such as the number of adversarial agents, their behavioral severity, and their network positions. By systematically

* Corresponding author at: School of Computer Science, University of Leeds, Leeds, UK.
E-mail addresses: ml21aa2a@leeds.ac.uk, aa.aldosery@uoh.edu.sa (A. Aldawsari), e.pournaras@leeds.ac.uk (E. Pournaras).

https://doi.org/10.1016/j.future.2025.108017
Received 25 January 2025; Received in revised form 18 May 2025; Accepted 7 July 2025
Available online 30 July 2025

examining these dynamics, the paper provides critical insights into the conditions under which systems transition from resilience to collapse, offering new insights for developing self-healing, fault-tolerant, and fault-correcting strategies in adversarial environments [20].

The main contributions of this paper are outlined as follows:

- An adversarial model for discrete-choice multi-objective optimization problems.
- A novel evaluation framework for characterizing resilience, vulnerability, and collapse in distributed optimization systems under adversarial influence.
- A comprehensive evaluation of the adversarial impact on system efficiency and agent discomfort.
- The first open large-scale benchmark datasets for discrete-choice adversarial optimization, generated from over 112 million experiments on real-world and synthetic inputs using the proposed adversarial model.
- New insights into how adversarial scale, severity, and structural positioning affect system optimality, including resilience and vulnerability thresholds, Pareto trade-offs, and structural vulnerabilities across diverse scenarios.
- An open-source software artifact implementing the proposed adversarial model within the I-EPOS system,[1] extending its functionality to support heterogeneous agent behaviors.

The rest of this paper is organized as follows: Section 2 reviews related work on adversarial distributed optimization. Section 3 introduces the proposed adversarial model and problem formulation, including the network model and optimization challenge. Section 4 outlines the experimental methodology and evaluation metrics. Section 5 presents the key findings, analyzing system behavior under varying adversarial conditions. Finally, Section 6 concludes the paper, discusses its limitations, and outlines directions for future research.

2. Related work

Resilience in distributed multi-objective optimization plays a critical role across domains such as smart grids, transportation, logistics, and communication networks, where robust and adaptive systems are crucial for ensuring operational efficiency [21,22]. Convex distributed optimization has received significant attention, with a focus on addressing challenges posed by adversary agents, network structures, and varied application domains [3,13,23]. Earlier work examined the robustness and vulnerability of consensus-based distributed optimization, focusing on addressing limitations related to adversary behavior, network topology, objective functions, and application domains [13,24,25]. The presence of adversary agents significantly impacts the performance of distributed optimization models. These agents disrupt optimization by slowing convergence, manipulating data, or withholding participation, resulting in suboptimal performance [8,15]. Table 1 provides a comparative analysis of related work on distributed optimization under adversarial conditions. It highlights key aspects such as the type of adversarial behavior[2], attack targets, system knowledge[3], network structures, and the impact on overall performance.

2.1. Adversary agents in distributed optimization

Yang et al. [3] provide a comprehensive survey on distributed optimization. Notable advancements include extensions of consensus-based protocols by Sundaram et al. [10] and Kuwaranancharoen et al. [26], which address adversarial threats in convex optimization. Su et al. [27] enhance these methods with decentralized architectures and explore adversarial influence on global objectives. However, these approaches assume adversary agents have full knowledge of the network topology and the private functions of all agents. This coordination among adversaries compromises the privacy of the agents in the system.

2.2. Adversarial attacks in multi-agent systems

Adversarial attacks significantly impact reinforcement learning (RL) systems across applications such as robotics, video games, and smart grids, undermining system stability and performance [40,41]. Lin et al. [28] demonstrate how adversarial perturbations affect cooperative multi-agent RL (c-MARL), showing its vulnerability compared to single-agent RL. Figura et al. [29] highlight how a single adversary can influence consensus-based c-MARL systems, disrupting team objectives. Zheng et al. [30] introduce criticality-based perturbations in deep Q-networks, demonstrating substantial performance degradation due to adversarial attacks. These studies show that an adversary can disrupt system operations and manipulate policies, influencing other agents to adopt behaviors aligned with its objectives.

2.3. Cyber-attacks and resilient control

Cyber-attacks, including data injection and denial-of-service (DoS) attacks, pose significant threats to distributed optimization by disrupting system operations and consensus mechanisms [31]. To address these challenges, Yemini et al. [32] introduce trust-based frameworks that mitigate malicious input, ensuring convergence to global optima. Similarly, Du et al. [33] and Zhao et al. [34] propose models relying on trusted agents to counteract adversarial influence. However, the effectiveness of these can be limited in scenarios with intermittent communication, such as ad hoc or robotic networks.

2.4. Resource allocation challenges under adversaries

In distributed resource allocation, adversarial disruptions are typically mitigated using robust optimization and detection mechanisms. Uribe et al. [35] and Turan et al. [36] propose primal–dual methods that tolerate Byzantine adversaries by identifying and eliminating malicious inputs, achieving resilience for up to 50% adversary density. Similarly, Ravi et al. [9] develop a detection method that uses agents' data values to identify and isolate potential malicious behavior, imposing an upper limit of 50% adversaries within the network topology. Gentz et al. [37] propose a detection method based on hypothesis-testing for insider attackers in randomized gossip-based sensor networks, leveraging statistical analysis of sensor states to identify malicious agents. While these methods enhance resilience against dispersed adversaries, they assume adversarial influence is evenly distributed and may not generalize to scenarios with concentrated or dynamic adversary placement.

2.5. Multi-objective distributed optimization

Recent studies have increasingly adopted Pareto-based multi-objective optimization to evaluate system trade-offs in complex infrastructures. Fettah et al. [42] introduce a Pareto strategy for optimizing distributed generation in power networks. Zhang et al. [43] formulate a multi-objective operational framework that integrates Pareto analysis to enhance resilience thresholds in distribution networks. Similarly, Boindala and Ostfeld [44] propose an optimization approach to balance

[1] Available at https://github.com/epournaras/EPOS.

[2] A malicious node sends the same value to all its neighbors at each time step, whereas a Byzantine node may send different values to different neighbors.

[3] Knowledge levels: full—complete knowledge of the network and agent objectives; partial—access to limited neighbor information; local—only own state or data is known.

Table 1
Comparison of literature on resilience in distributed optimization.

		[10]	[26]	[27]	[28]	[29]	[30]	[31]	[32]	[33]	[34]	[35,36]	[9]	[37]	[38,39]	This work
Adversarial behavior	Malicious	✓			✓	✓			✓	✓	✓		✓	✓		✓
	Byzantine	✓	✓	✓						✓	✓					
	Cyber-attacks						✓	✓	✓		✓					
	Eavesdropping													✓		
Adversary target	Consensus	✓	✓			✓		✓			✓		✓	✓		
	Information Exchange			✓					✓		✓			✓		
	System Objective						✓		✓							✓
	Observation				✓											
Knowledge of the system	Full	✓	✓	✓			✓	✓							✓	
	Partial					✓		✓						✓		
	Local										✓	✓			✓	
Directed Network			✓	✓	✓					✓		✓	✓			✓
Performance measure	Convergence		✓	✓				✓	✓	✓	✓	✓	✓	✓	✓	
	Distance to Optimality	✓	✓	✓											✓	
	Reward/Utility				✓	✓	✓				✓					
	Efficiency															✓
Algorithms / Techniques for Optimization		Local filtering	Distance–based filtering	Local filtering	Gradient –based (Deep Q-learning)	Consensus-based MARL	None	Mean subsequence reduced	Probabilistic trust– based & projection–based	Markov switching topology & Push-DIGing	Resilience with trusted agents & dominating set	Primal–Dual	FROST	Randomized gossip	Differentially private gradient tracking	I-EPOS

reliability, cost, and failure risk using Pareto fronts. While these studies underscore the value of Pareto analysis for resilient optimization, they focus on centralized infrastructures and do not address adversarial influence or the complexities of distributed, multi-agent decision-making explored in this work.

2.6. Privacy-preserving distributed optimization

Privacy-preserving distributed optimization safeguards sensitive agent information against eavesdropping adversaries using techniques such as differential privacy [38,39,45], homomorphic cryptography [46,47], and gradient perturbation [48,49], to ensure secure information exchange. However, these studies focus on privacy protection rather than adversarial behavior in optimization contexts.

2.7. Combinatorial optimization algorithms

In distributed combinatorial optimization, Hinrichs et al. [50,51] propose COHDA, a combinatorial optimization heuristic designed for multi-agent systems. However, COHDA encounters scalability challenges due to increasing communication overhead as the number of agents grows. The collective learning approach of Pournaras et al. [52] address this with EPOS (Economic Planning and Optimized Selections), a distributed optimization method that enables agents to collaboratively optimize global resource allocation, particularly in participatory sharing economies. Although EPOS ensures privacy, autonomy, and scalability, it faces computational limitations when applied to wide tree structures with multiple child nodes [53]. I-EPOS is the iterative extension of EPOS; it incorporates decentralized iterative back-propagation and localized decision-making to enhance scalability and support plan coordination across deeper and broader network hierarchies [53,54].

While COHDA, EPOS, and I-EPOS represent foundational combinatorial optimization approaches, their system performance under adversarial conditions has not been studied before, despite some limited work on measuring the impact of arbitrary structural faults, which does not focus on agents' behavior [55]. In this work, we introduce a generic adversarial model applicable to such settings. The model enables a structured evaluation of resilience, vulnerability, and collapse dynamics, and, to the best of our knowledge, is the first to systematically explore adversarial behaviors in discrete-choice combinatorial optimization.

A large body of research to date focuses on continuous distributed optimization, often assuming limited adversary presence or relying on complete graph topologies [8–10,13]. Such assumptions do not fully capture the complexity of real-world systems, where distributed structures, heterogeneous agent behaviors, and dynamic adversarial threats are prevalent [10,30]. Although existing solutions offer valuable mitigation strategies [33,35], the lack of comprehensive analyses on inherent system vulnerability, resilience thresholds, and pathways to optimization collapse remains a gap.

To fill this gap, we propose a generic adversarial model to systematically analyze how adversarial agents influence system performance and stability in distributed multi-objective optimization. Our objective is to evaluate how adversarial behavior influences resilience, vulnerability, and collapse, while informing the development of self-healing strategies for robust optimization. Moreover, his work releases the first large-scale, open benchmark dataset designed for evaluating adversarial impacts under discrete-choice optimization settings. To clarify the novelty of our contribution, Table 2 summarizes a comparative analysis with existing work, highlighting key aspects such as discrete decision-making, multi-objective formulation, resilience thresholds, structural vulnerability, and Pareto-based evaluation.

3. Adversarial distributed optimization

3.1. Problem formulation

Resilience in distributed optimization is essential for maintaining system performance under adversarial conditions. Adversary agents disrupt operations, degrade efficiency, and increase vulnerability. This raises key questions: How do adversary agents influence the efficiency and stability of distributed optimization systems? What thresholds of adversarial behavior lead to transitions from resilience to vulnerability or collapse? How do parameters such as adversary density, adversarial severity, and network positioning influence these transitions?

To address these challenges, we propose a generic adversarial distributed optimization model tailored to discrete-choice scenarios. Our model investigates the trade-offs between system-wide and individual agent goals in adversarial settings. It incorporates key parameters, including the scale of adversaries, behavioral severity, and structural positioning, providing a robust framework to evaluate system vulnerability and resilience. Through this study, we identify critical thresholds

Table 2
Comparison of the novelty aspects of this work with related distributed optimization approaches.

Criteria	This work	[10]	[26]	[27]	[28]	[29]	[30]	[31]	[32–34]	[35,36]	[9]	[37]	[38,39]	[42–44]
Multi-objective optimization	✓			✓		✓			✓	✓	✓		✓	✓
Discrete decision-making	✓				✓	✓	✓							
Resilience & vulnerability thresholds	✓	✓	✓	✓				✓	✓	✓	✓	✓		✓
Structure analysis	✓	✓		✓				✓	✓					
Pareto analysis	✓													✓

Table 3
Nomenclature utilized in this research.

Notation	Description
A	set of agents in the network
$\mathsf{n}= \lvert A \rvert$	number of agents
$A_{\mathsf{d}} \subseteq A$	set of adversary agents
$A_{\mathsf{l}} \subseteq A$	set of legitimate agents
P_{a}	set of possible plans of agent a
$p_{\mathsf{a,i}} \in P_{\mathsf{a}}$	plan i of agent a
k	number of plans
d	size of plan
s_{a}	selected plan of agent a
g	global response
D	discomfort cost
f_D	discomfort cost function
I	inefficiency cost
f_I	inefficiency cost function

for transitions from resilience to collapse, providing an in-depth understanding of system behavior under adversarial influence. These findings inform the development of self-healing strategies that enhance fault-tolerance and mitigate adversarial impacts across diverse distributed optimization applications.

3.2. Network model

Consider a network with n agents, denoted by A, each identified by a unique ID in the set $\{1, 2, 3, \ldots, \mathsf{n}\}$. The network topology is represented as a connected graph $G = (A, E)$, where A is the set of agents, and E is the set of edges, with $(j, i) \in E$ not necessarily implying $(i, j) \in E$. Agents interact within a self-organized network through bidirectional communication to exchange information and update their states to align with system goals.

Table 3 summarizes the notations used throughout the paper to formalize the network model and optimization framework.

3.3. Optimization framework in discrete-choice scenarios

In distributed discrete-choice optimization, each agent $a \in A$ selects one option from a finite set of k alternatives referred to as *possible plans* $P_{\mathsf{a}} \subset \mathbb{R}^d$. Each plan $p_{\mathsf{a,i}} \in P_{\mathsf{a}}$ is a sequence of size d that represents a decision configuration, such as resource allocation or scheduling. These plans reflect the agent's potential future operations, from which the agent selects one, denoted as s_{a}. The collective outcome is captured by the *global response* $g = \sum_{a \in A} s_{\mathsf{a}}$, which aggregates all selected plans of all agents to evaluate system-level performance. For instance, in power grid systems, each household acts as an agent with multiple plans representing alternative appliance energy consumption levels [56]. Each household selects one plan, contributing to the total energy consumption, which represents the global response (g) in power grid systems.

Agents aim to balance their individual preferences with system-wide goals, which often involve conflicting criteria. Each agent a has an individual preference for its plans, quantified by the *discomfort cost* (D), such that $D_{a,i} = f_D(p_{\mathsf{a,i}})$, where f_D measures how undesirable plan i is for agent a based on the agent's preferences; lower costs indicate more preferred plans. Each agent a evaluates the costs D for each possible plan $p_{\mathsf{a,i}} \in P_{\mathsf{a}}$ and the plan with the minimum cost is selected.

While agents aim to minimize their own discomfort, they may also consider system-wide metrics such as the *inefficiency cost* (I). The inefficiency cost (I) is the measure used to evaluate the collective system-wide performance based on the aggregated responses of all agents. It represents the system-wide performance inefficiency that agents aim to minimize through coordinated decision-making: $I = f_I\left(\sum_{a=1}^{\mathsf{n}}(s_{\mathsf{a}})\right)$. Each agent selects a plan that minimizes a weighted combination of individual discomfort and system-wide inefficiency, as shown in Eq. (1).

$$\begin{aligned} s_{\mathsf{a}} &= \arg\min_{i=1}^{k} \left(\alpha_a \cdot I_{a,i} + \beta_a \cdot D_{a,i}\right) \\ &= \arg\min_{i=1}^{k} \left[\alpha_a \cdot f_I\left(s_1 + s_2 + \cdots + s_{\mathsf{n}}\right)\right. \\ &\quad \left. + \beta_a \cdot f_D\left(D_{1,s}, D_{2,s}, \ldots, D_{\mathsf{n},s}\right)\right], \end{aligned} \tag{1}$$

where $\alpha_a + \beta_a = 1 \quad \& \quad \alpha_a, \beta_a \in [0, 1]$

The behavior of agent a is modeled by the corresponding weights α_a and β_a, which represent agent's priorities between minimizing system-wide inefficiency and personal discomfort, respectively. A higher weight indicates a higher preference for minimizing the corresponding objective. On the other hand, a weight of 0 means that the corresponding objective is not considered. For instance, an agent with $\alpha_a = 1$ and $\beta_a = 0$ behaves altruistically, prioritizing global goals. Conversely, $\alpha_a = 0$ and $\beta_a = 1$ define a selfish agent focusing solely on its individual preference.

3.4. Adversarial distributed optimization model

We propose an adversarial model applicable across a range of combinatorial optimization scenarios. In this model, the agent population A is partitioned into two disjoint subsets: legitimate agents A_{l} and adversary agents A_{d}, such that $A = A_{\mathsf{l}} \cup A_{\mathsf{d}}$. While legitimate agents, $A_{\mathsf{l}} \subseteq A$, align their actions with system-wide objectives to optimize overall performance, adversary agents, $A_{\mathsf{d}} \subseteq A$, prioritize their individual interests over collective system goals by adapting their behavior to maximize personal benefits. For instance, in a bike-sharing system [53], optimization ensures a balanced distribution of bikes across stations to meet user demand. Legitimate users may select pick-up and drop-off stations while considering system-wide efficiency, maintaining network equilibrium. In contrast, adversary users prioritize their own convenience, selecting stations solely based on personal preference, leading to imbalances such as empty or overloaded stations, and ultimately degrading overall efficiency and user satisfaction.

Adversarial behavior is modeled by adjusting the weight β_a in the agent's decision function (Eq. (1)). Legitimate agents are assigned $\beta_l = 0$, fully aligning with system goals, while adversary agents are assigned $\beta_d > 0$, increasing emphasis on personal discomfort minimization at the expense of system-wide efficiency. This parameterization allows

adversarial behavior to be modeled in a continuous space, from fully altruistic to fully selfish.

By varying the distribution and severity of adversarial weights across the agent population, our model enables systematic analysis of resilience, vulnerability, and collapse in distributed optimization. This includes assessing how the impact of adversarial behavior is linked to the network positioning of the adversary agents and legitimate agents. Adversarial behavior amplifies the inefficiency cost I, reflecting the trade-off between minimizing individual discomfort (D) and optimizing overall system efficiency (I). While D focuses on individual preferences, I addresses the system-wide inefficiency caused by deviations from optimal resource allocation, underscoring the conflict between individual and collective optimization goals.

4. Experimental methodology

This section illustrates the distributed optimization method employed as a case study, the experimental setup, the application scenarios, the measured variables and the evaluation metrics.

4.1. Distributed optimization method

The adversarial distributed optimization model is implemented within the *Iterative Economic Planning and Optimized Selections* (I-EPOS) framework. I-EPOS is a discrete-choice distributed combinatorial optimization algorithm for large-scale multi-agent networks [53,54]. It employs a self-organized, multi-level hierarchical structure to enable efficient communication, coordination, and scalability while minimizing communication overhead [57].

I-EPOS enables the agents to iteratively coordinate their choices in collective decision-making. Each iteration consists of two distinct phases: a bottom-up phase and a top-down phase. During the bottom-up phase, agents select plans based on the aggregated choices of agents in the branch beneath, as well as the selections made by all agents in the previous iteration. Conversely, the top-down phase addresses incomplete knowledge from higher branches in the hierarchy, enabling agents to revert to previous selections if no cost reduction is achieved. This process continues until a predefined iteration limit is reached or no further improvement in the optimization objective occurs [53,54]. This iterative coordination mechanism addresses the inherent complexity of multi-agent optimization, particularly under non-linear cost functions and incomplete knowledge of other agents' choices. These conditions make the optimization problem NP-hard [53], requiring distributed coordination methods that allow agents to refine their decisions based on both local preferences and system-wide impact.

The hierarchical network is structured as an acyclic graph, where each parent agent aggregates responses from its children by avoiding double counting. This design ensures efficient coordination when optimizing individual decisions and system-wide objectives [53,55,57]. I-EPOS is well-suited for adversarial scenarios in large-scale distributed systems due to its scalability, adaptability to diverse agent behaviors, and potential mitigate adversarial conditions [54,58].

To apply the proposed adversarial model, the I-EPOS framework was extended to support heterogeneous agent behaviors. The original implementation assumed uniform agent preferences across the population. We enhanced the framework to allow agents to configure individual decision-making weights. This enhancement enables modeling adversarial agents with varying levels of behavioral severity and is made available as an open-source artifact to support reproducibility and future research[4].

[4] Available at https://github.com/epournaras/EPOS.

4.2. Experimental setup

Experiments are conducted using multiple HPC servers with varying configurations that support large-scale experimentation and ensure computational efficiency. These include high-memory nodes (up to 768 GB) and multi-core processors (up to 40 cores per node). In addition to these servers, the University of Leeds ARC4 system[5] is utilized. The ARC4 system includes two nodes, each equipped with 40 cores, 768 GB of memory, and 800 GB of storage, providing robust computational capacity for large-scale experiments.

4.3. Application scenarios

Adversarial distributed optimization is studied in three application scenarios based on real-world data and a synthetic dataset. Table 4 provides an overview of the datasets used in this research, including the agent populations, the number and size of plans per agent, agent representation and the interpretation of discomfort and inefficiency costs within each application domain. These costs are defined explicitly through the optimization objectives and reflect domain-specific constraints. Further mathematical definitions of cost functions are provided in Appendix A.

4.3.1. Energy-demand dataset

The energy application scenario uses a dataset derived from simulated zonal power transmission in the Pacific Northwest.[6] The dataset includes power consumption profiles for 1,000 users, with each user represented by an agent containing 10 possible plans. Each plan comprises a 144-length sequence representing electricity consumption at 5 min intervals over a 12-hour period. These plans are generated using load-shifting strategies to balance grid load during peak and off-peak hours, reducing strain on the energy system. Plans are ranked by preference scores ranging from 0 to 1, with higher scores reflecting greater alignment with the user's original consumption patterns. The inefficiency cost is measured as the deviation in aggregated energy consumption from the desired load-balancing levels, capturing the system ability to maintain stability and efficiency. Adversarial households disrupt the system by selecting plans that counteract load balancing, thereby increasing the risk of grid instability during peak periods.

4.3.2. Privacy dataset

The privacy dataset originates from a living-lab experiment at the Decision Science Laboratory[7] of ETH Zurich, involving 72 participants evaluating 64 data-sharing scenarios that involve 4 sensor types, data collectors, and contexts [58]. The data-sharing choices of each participant in the experiment determine three data-sharing plans, representing their intrinsic motivation to share and two rewarded scenarios. The plans are assessed using privacy valuation schemes assigning normalized costs ranging from 0 to 1, where lower costs indicate less privacy compromise. The dataset facilitates testing under high and low privacy-preservation target signals. The inefficiency cost is calculated by the residual sum of squares between the shared and the desired data that measures their mismatch and is an indicator of quality of service supported by the collected data. Adversarial participants disrupt coordination by focusing solely on minimizing their data sharing, under-mining the quality of service of data collectors.

[5] ARC4 is an HPC cluster at Leeds providing a Linux-based HPC service based on CentOS 7. More information: https://arcdocs.leeds.ac.uk/systems/arc4.html.

[6] Available upon request at http://www.pnwsmartgrid.org/participants.asp.

[7] https://www.descil.ethz.ch

Table 4
Description of the datasets and experimental setup in this research.

Dataset name	No. agents	No. plans	Plan size	Agents	Discomfort cost	Inefficiency cost	Total experiments
Energy	1000	10	144	Households	Time shift from intrinsic preference	Variance of energy demand	3,118,560
Privacy	72	3	64	Smart phone users	Privacy loss	Mismatch between shared and desired data	498,780 (2 target signals)
Voting	266	31	5	Voters	Compromise distance from intrinsic voting preferences	Polarization	103,456,800 (120 target signals)
Gaussian (synthetic)	10–100	2–10	2	Simulated agents	Ranking distance	Variance	4,125,000

4.3.3. Voting dataset

This new dataset is derived from voting data in a regional election with five candidates and 266 voters[8] [59]. Each voter has 31 alternative voting plans, representing ranked preferences among the five candidates. The optimization focuses on minimizing polarization in the voting outcomes, which refers to reaching the same voting outcome but with compromises that reduce polarization. Polarization here is the inefficiency cost and it refers to the mismatch from a liner ranking of the alternatives in the voting outcomes, although other polarization models could be studied as well [60]. The rationale of linearity is to deviate from concentrating the voters' preferences to two opposing poles. To control for the same voting outcome, 120 target signals are generated from all combinations of values 0, 0.25, 0.5, 0.75, and 1, representing the linear ranking of alternatives. Adversarial behavior occurs when voters prioritize their intrinsic preferences, i.e., no compromises to reduce polarization.

4.3.4. Synthetic Gaussian dataset

The synthetic dataset is constructed to evaluate system scalability under controlled, domain-agnostic conditions. It includes 100 agents with 10 generated plans, where each plan is a 100-dimensional vector sampled from a Gaussian distribution $\mathcal{N}(0,1)$. Plans are sorted by their index, with lower indices arbitrary indicating higher agent preference. Discomfort cost is defined as the rank of the selected plan—i.e., a higher index reflects greater deviation from the most preferred option. Inefficiency cost is measured as the variance of the aggregated global response, capturing system-wide imbalance. This synthetic setup allows systematic analysis of adversarial effects across varying agent populations, number/size of plans, and attack configurations.

4.4. Varying dimensions and performed experiments

The following dimensions are studied in the performed experiments:

- **Scale of adversaries** ($|A_{\mathrm{d}}|$): Incrementally increase the number of adversary agents A_{d} from 1 to n across all datasets to analyze performance under varying adversary densities; i.e., $A_{\mathrm{d}} = \{a \mid a \in A\}$.
- **Adversarial severity** (β): The adversarial preference to minimize discomfort cost (β) is varied across 30 levels, with β incrementing from 0 to 1 such that $\beta = \frac{b}{30}$ for $b = \{1, 2, 3, \ldots, 30\}$.
- **Adversary position:** The influence of adversary positions within the hierarchical network is analyzed using two approaches: layer-wise and cumulative structural analysis. A binary tree structure is employed, with each hierarchical layer containing approximately $\log_2 |A|$ agents, where $|A|$ is the total number of agents. The structural analysis evaluates inefficiency costs under varying adversary scales (25%, 50%, 75%, 100%) at each layer of the hierarchy. The cumulative analysis examines the aggregated impact of adversary agents positioned incrementally in top-down (root-to-leaf) and bottom-up (leaf-to-root) configurations.

For each dataset, experiments are conducted across 30 adversarial severity levels (β) with 100 simulation runs per configuration.

Layer-wise structural analysis is performed at four adversarial proportions $p \in \{25\%, 50\%, 75\%, 100\%\}$ within each layer of the hierarchical topology, where the number of layers is defined as $\lceil \log_2 |A| \rceil$. For each layer L, the number of adversarial agents is calculated as $k_p = \max\left(1, \left\lfloor \frac{p}{100} \cdot |A_L| \right\rfloor\right)$, where $|A_L|$ denotes the number of agents in layer L. Adversarial configurations are sampled up to 100 combinations per setting to ensure computational feasibility.[9] In addition, two cumulative structural attack scenarios are simulated, top-down (root-to-leaf) and bottom-up (leaf-to-root), introducing $2 \times |A|$ further experiments per dataset. The total number of experiments per dataset is calculated as:

$$\text{Total Experiments} = (30 \times \text{number of signals}) \times \left[(100 \times |A|) + \sum_L \sum_p \min\left(100, \binom{|A_L|}{k_p}\right) + (2 \times |A|)\right]$$

For the synthetic Gaussian dataset, experiments vary both agent populations and the number of plans per agent (from 2 to 10). The total number of experiments is computed as: $\text{Total Experiments}_{\text{Gaussian}} = \sum_{i=1}^{10} (10i \times 30 \times 5 \times 50)$, where i is the number of agents, 30 is the number of severity levels, 5 is the number of plans, and 50 is the number of random structural permutations (i.e., reordering of agents in the tree).

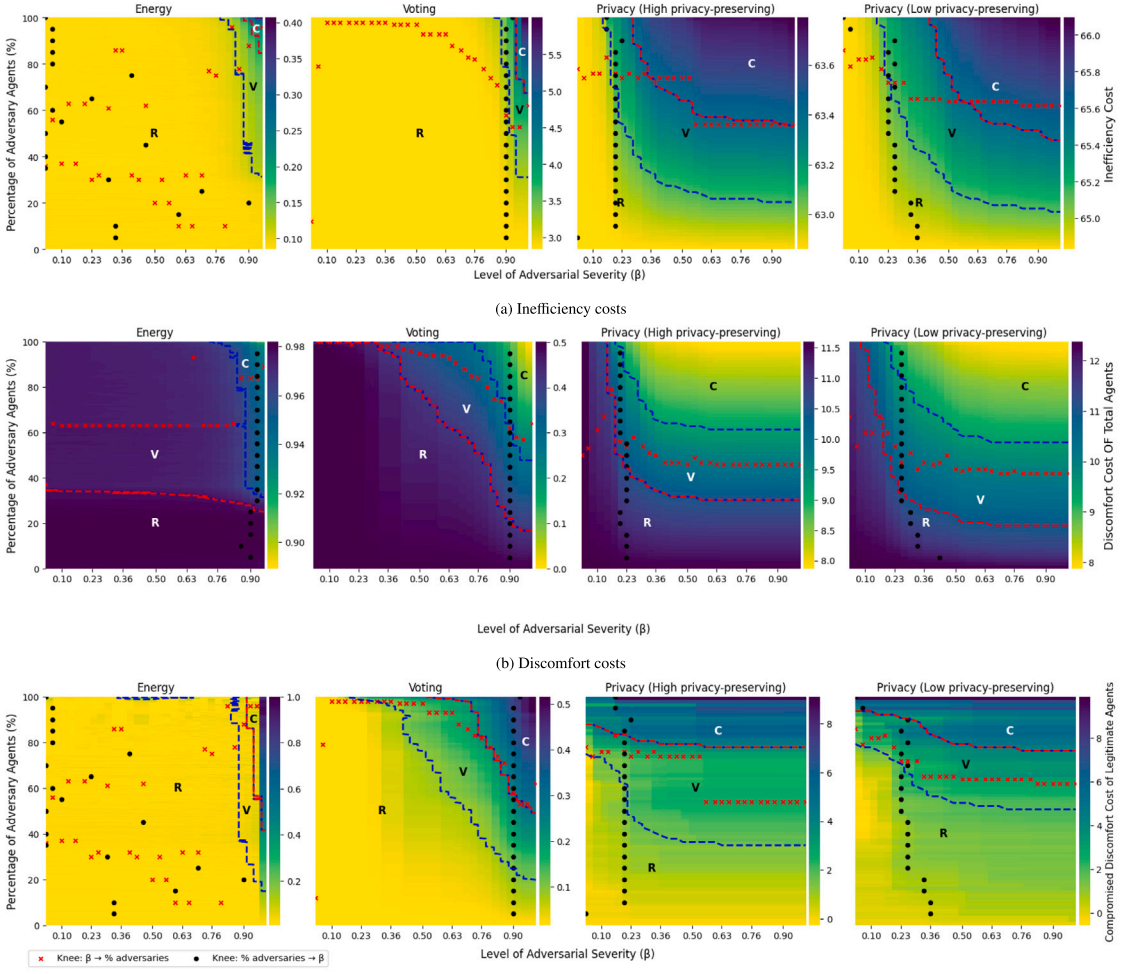
4.5. Evaluation metrics

Optimization objectives: System performance is assessed using three metrics: inefficiency cost I, discomfort cost D, and the compromise cost of legitimate agents. Inefficiency cost captures the deviation from optimal system performance. Discomfort cost reflects individual agents' dissatisfaction, i.e. to what extent a plan is not the most preferred one. The compromise cost quantifies the increase in discomfort experienced by legitimate agents due to adversarial influence. It is calculated as the difference in discomfort between scenarios with and without adversaries, highlighting the collective burden legitimate agents bear to maintain system performance.

Pareto optimality: In multi-objective optimization, the Pareto front defines solutions where no objective can improve without compromising another objective. The knee point on this front identifies the most balanced trade-off between competing objectives. This study uses the Minimum Manhattan Distance (MMD) method to locate the knee point, measuring the distance from each Pareto solution to an ideal reference point where both objectives are optimized. The solution with

[8] The UK Labor Party Leadership Vote Available at https://preflib.simonrey.fr/datasets.

[9] The equation assumes up to 100 combinations per layer-percentage, but the binary hierarchy often yields fewer due to limited agents per layer.



(a) Inefficiency costs

(b) Discomfort costs

(c) Compromised discomfort costs for legitimate agents

Fig. 1. Inefficiency, discomfort, and compromised discomfort over adversary scales and severity in the energy, voting, and privacy datasets, including Pareto knee points and resilience (R), vulnerability (V), and collapse (C) thresholds.

the smallest distance is selected. This approach ensures a balanced trade-off between discomfort (D) and inefficiency (I) in line with established approaches [61,62].

Resilience, vulnerability, and collapse framework: To classify system states, the multi-Otsu thresholding method is applied to segment inefficiency and discomfort values into three distinct regions: resilience (low inefficiency), vulnerability (moderate inefficiency), and collapse (high inefficiency). This technique minimizes intra-class variance, offering a robust framework to detect transitions in system performance under adversarial conditions [63,64].

5. Results analysis and discussion

This section presents the results of extensive experiments evaluating the impact of adversarial agents on multi-objective distributed optimization across the real-world (energy, voting, and privacy) and synthetic datasets.

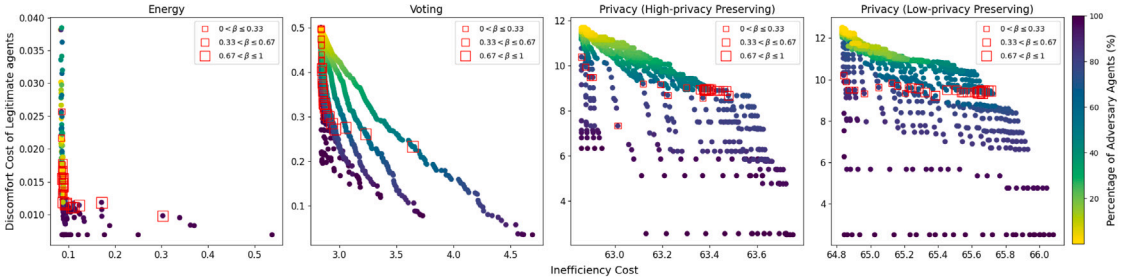
5.1. Resilience analysis

Fig. 1 presents inefficiency, discomfort, and compromised discomfort costs across varying adversary scales and severity levels (β). These metrics capture degradation in system performance, agent satisfaction, and the impact on legitimate agents. The graphs include resilience, vulnerability, and collapse thresholds, with overlaid Pareto knee points. For the inefficiency and compromised-discomfort plots, knees mark the best trade-off between those two metrics; on the discomfort heatmaps, the knee indicates the minimal total discomfort for a given inefficiency.

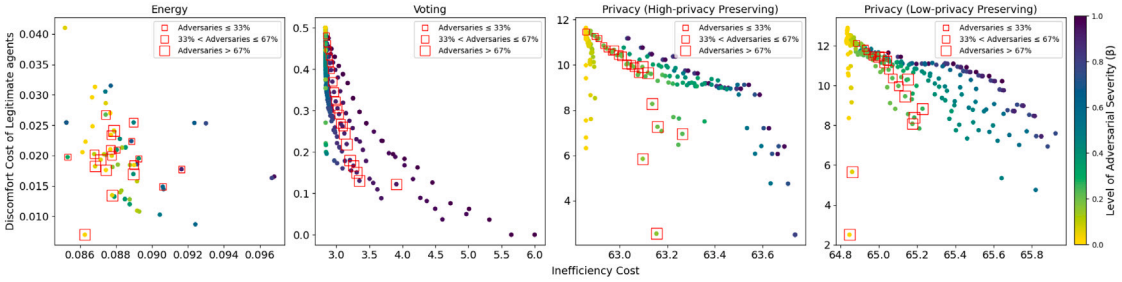
Inefficiency cost (Fig. 1(a)) remains low in the resilience zone, especially when adversary ratios are below 30%–50% and $\beta < 0.8$. As adversary scale and severity increase, systems shift from Resilience to Vulnerability and Collapse, with thresholds varying across datasets. Energy and voting maintain 90% resilience and only 3%–5% collapse, while privacy configurations show earlier collapse near 20%. In energy, inefficiency peaks at $\beta = 1$ near 4000 cost,[10] when adversaries exceed 85%. In the voting dataset, inefficiency increases by 111% in the vulnerable region, with collapse triggered beyond 70% adversaries at $\beta \geq 0.96$. Privacy collapses earlier: $\beta \geq 0.3$ at 50% adversaries in the high privacy-preserving signal with 50% adversaries, and $\beta \geq 0.5$ in the low privacy-preserving signal.

Discomfort cost (Fig. 1(b)) shows a consistent decline as adversarial intensity increases. In resilient regions, typically below 30%–50% adversary presence and $\beta < 0.7$, discomfort initially remains high but declines gradually, then sharply in collapse regions, as adversary scale and severity increase. In the energy dataset, discomfort remains high only below 30% adversaries and drops by 8% before collapsing to near-zero. The voting dataset follows a similar pattern, with discomfort gradually decreasing and fully eliminated in collapse. The privacy

[10] Values for $\beta = 1$ in the energy dataset are excluded from visualizations to avoid heatmap saturation due to extremely high inefficiency values.



(a) Pareto front and knee points across adversarial severity.



(b) Pareto front and knee points across scales of adversaries.

Fig. 2. The pareto optimality of the energy voting, and privacy datasets.

datasets exhibit earlier collapse ($\beta > 0.3$, >60% adversaries), resulting in faster discomfort decline.

Compromised discomfort cost (Fig. 1(c)) increases with adversarial presence. In the energy dataset, it remains low for $\beta < 0.8$ even when all agents are adversarial, but rises sharply when the adversary scale exceeds 40% at $\beta > 0.9$. In voting, the cost remain low under mild attacks ($\beta < 0.2$, adversary scales < 20%) and increase gradually beyond 50% at $\beta > 0.7$. Privacy datasets show a steeper rise, peaking at 9 (high privacy) and 10 (low privacy) when 90% of agents are adversarial. Resilience disappears at high adversary densities (> 80%), even under low severity, and collapse emerges at low β when agent compromise is widespread. While energy and voting are more sensitive to β, privacy is primarily affected by adversary scale.

Interestingly, a noticeable resilience lag exists between the system-level inefficiency and the agent-level discomfort and compromise costs. In all datasets, a substantial portion of configurations classified as "Vulnerable" or even "Collapsed" by inefficiency remain in the "Resilient" state when evaluated by discomfort metrics. For instance, in the energy dataset, while only 6% of configurations are in vulnerability based on inefficiency, 59% fall under vulnerability based on discomfort—indicating that discomfort degrades far earlier. Similarly, in the voting dataset, 21% of configurations are classified as vulnerable by inefficiency, compared to 70% by discomfort. This pattern reveals a lag of up to 40%–60% in the transition from individual to system-level degradation, with discomfort and compromise metrics acting as early-warning indicators long before global inefficiency surfaces. This insight underlines the importance of incorporating agent-centric metrics for proactive resilience monitoring.

5.2. Pareto optimality analysis

Pareto optimality analysis identifies fronts and knee points reflecting optimal trade-offs under varying adversarial conditions. Fig. 2 shows how system inefficiency relates to discomfort experienced by legitimate agents across different severity levels and adversary scales.

Fig. 2(a) focuses on the impact of adversarial severity (β) on the tolerated scale of adversaries. The voting dataset maintains stable knee points across 50%–60% adversary ratios over a broad severity range ($0.1 \leq \beta \leq 0.7$), with consistent fronts even at high β. The privacy dataset shows distinct patterns: the high privacy-preserving signal tolerates 70% adversaries at $\beta < 0.5$, decreasing to 50% at higher severities; the low privacy-preserving signal exhibits smoother transitions with average tolerance near 68%.

Fig. 2(b) evaluates the impact of adversarial scale on tolerated severity. For visual clarity, only 20 adversary scales are shown per dataset. In energy, knee points are stable at $0.03 < \beta < 0.3$ for up to 90% adversaries, increase to $\beta = 0.9$ at lower scales (< 40%), and decline to $\beta < 0.1$ at full scale. Voting shows higher tolerance, with knee points extending $\beta = 0.8$ even under full adversaries. Privacy dataset shows greater variability: the high privacy-preserving configuration reaches $\beta = 0.2$ for 50%–90% adversaries, while the low privacy-preserving case peaks at $\beta = 0.36$ for 20% adversaries and declines to $\beta = 0.13$ at full scale. Detailed Pareto plots are in Appendix C.

5.3. Structure analysis

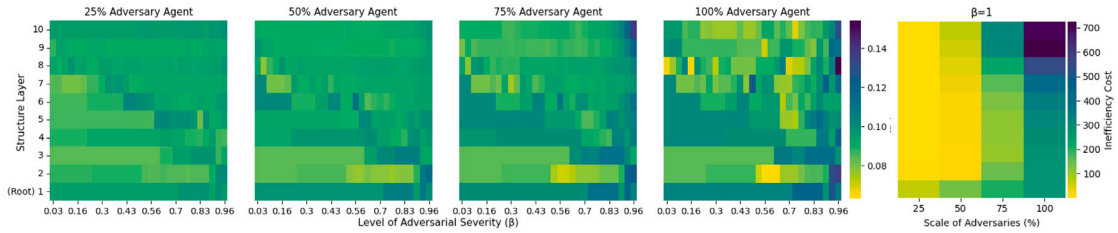
This section analyzes how hierarchical structures influence inefficiency costs across datasets using two approaches: layer-wise and cumulative structural analysis.

5.3.1. Layer-wise structural analysis

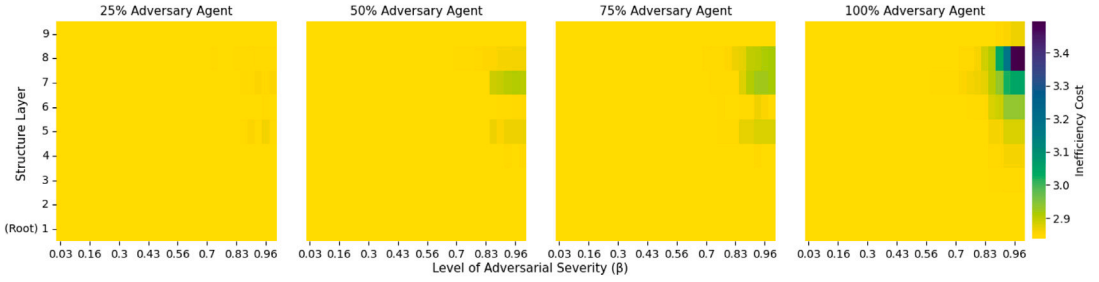
Fig. 3 presents the inefficiency costs across hierarchical layers under adversary scales of 25%, 50%, 75%, and 100%. For layers with few agents, intermediate adversary ratios are approximated by averaging to the closest feasible configurations.

In the energy dataset (Fig. 3(a)), with 1,000 agents over 10 layers, inefficiency remains low (below 0.16) across moderate adversarial scales and gradually increases with severity ($\beta < 1.0$). The minimum inefficiency occurs at layer 2 under 100% adversaries and $\beta = 0.56$, while inefficiency peaks at layer 8 under $\beta = 0.96$, marking a 144% increase from the minimum. Profiles remain smooth under 25% and 50% adversary scales but fluctuate more at 75% and 100%. Under extreme severity ($\beta = 1$), inefficiency surges sharply, reaching over 723 at full adversarial saturation. With this severity, the root layer shows moderate inefficiency, exceeding layers 2–7 at 75% and matching layers 2 and 3 at 100%, followed by a steady increase down the hierarchy. Despite layer 10 hosting the largest agent population, its inefficiency values are slightly lower than those of layer 9.

In the voting dataset (Fig. 3(b)), with 266 agents over 9 layers, inefficiency remains stable across all layers under low adversarial severity ($\beta \leq 0.3$). As severity increases, costs escalate, particularly at the root



(a) Energy dataset.



(b) Voting dataset.

(c) Privacy dataset (High privacy-preserving).

(d) Privacy dataset (Low privacy-preserving).

Fig. 3. Inefficiency costs across hierarchical structure layers under various adversarial configurations.

(Layer 1) and Layers 2 and 3, which consistently experience higher inefficiency than deeper layers. The maximum inefficiency is observed at layer 8 under 100% adversaries, where agent density is highest. Layer 9,despite its depth, shows lower inefficiency due to a smaller agent count. At 25% and 50% adversary scales, inefficiency profiles remain relatively smooth; however, at 75% and 100%, cost escalations become pronounced, particularly in upper and middle layers.

In the privacy dataset (Figs. 3(c) and 3(d)), with 72 agents across 7 layers, inefficiency remains low and uniform under low severities. In the high privacy-preserving configuration, the root layer consistently incurs higher costs than subsequent layers across all adversary scales, with inefficiency peaking at layer 6 (the layer with the largest number of agents) under $\beta \geq 0.7$. Layer 7, although deeper, has lower costs due to fewer agents. In the low privacy-preserving configuration, similar trends are observed with occasional fluctuations between layers 5 and 6 at higher adversary scales. Overall, inefficiency profiles remain structurally consistent between both privacy settings, though severity levels accelerate cost increases in the high-privacy case.

5.3.2. *Cumulative structural analysis*

Cumulative analysis evaluates how adversarial influence propagates through hierarchical structures in two configurations: top-down (root-to-leaf) and bottom-up (leaf-to-root). Fig. 4 shows inefficiency across layers under both directions.

In the top-down positioning, the energy dataset remains resilient up to layer 7 with 20% adversaries. Collapse occurs at layer 10 when the adversary ratio exceeds 50% and $\beta > 0.9$. In voting, resilience holds through layer 4, with vulnerability at layer 5 and collapse at layers 6–7 under high severity ($\beta > 0.9$). In the privacy datasets, both signals remain resilient in the top four layers, with vulnerability emerging at layer 4 for $\beta > 0.7$. Collapse appears at layer 6 in the high privacy-preserving signal with over 60% adversaries, and slightly earlier in the low privacy-preserving signal with 50% adversaries.

In the bottom-up direction, the energy dataset shows a narrower vulnerability and collapse region. Vulnerability begins at layer 9 with 50% adversaries, and collapse follows at layer 10 under high severity

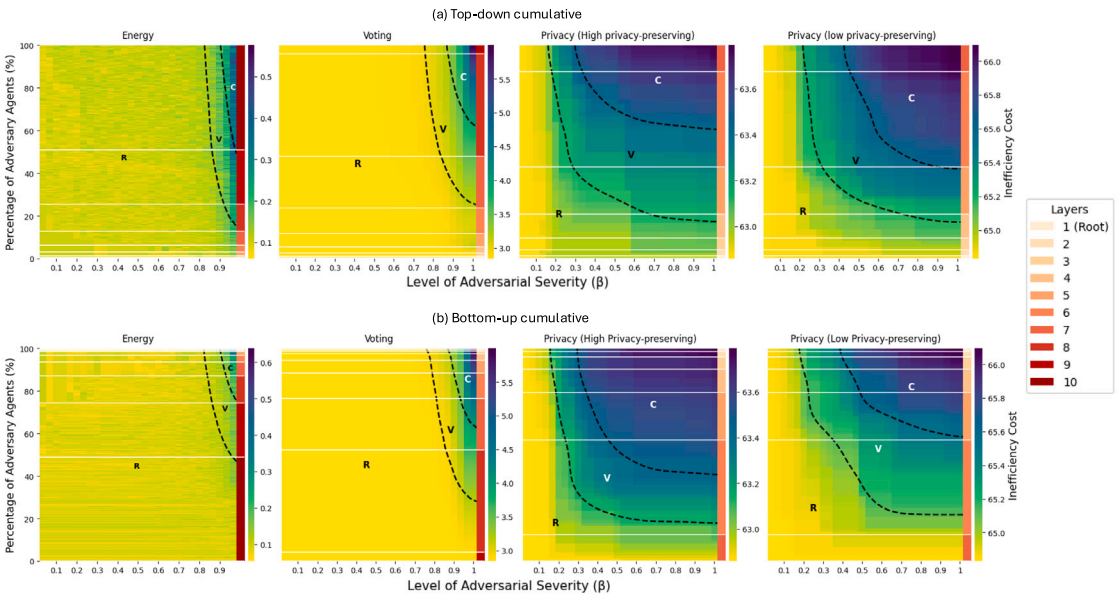


Fig. 4. Inefficiency cost across hierarchical structure layers under various adversarial configurations in energy, voting and privacy datasets.

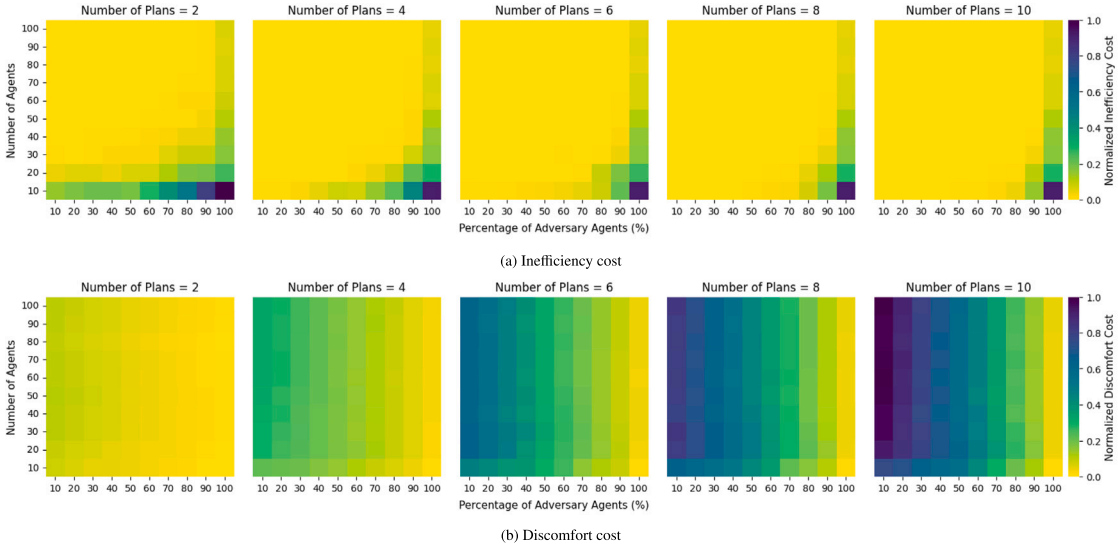


Fig. 5. Normalized inefficiency and discomfort costs on synthetic Gaussian data across varying numbers of agents, plan options per agent, and adversary ratios.

($\beta > 0.9$). Voting results are similar across both directions, with transitions driven more by adversary ratio than structural depth. Vulnerability emerges at layer 6 with 20% adversaries, and collapse follows at layer 5 with 60%. In privacy, vulnerability appears at layer 6 in both signals; collapse occurs at layer 6 (high privacy-preserving) and layer 5 (low privacy-preserving).

5.4. Scalability analysis with synthetic data

To benchmark scalability, we use a synthetic Gaussian dataset to systematically vary agent population (10–100) and plan count (2–10) under different adversarial ratios. This controlled setup reveals optimization performance across configurations under adversarial pressure. Fig. 5(a) shows normalized inefficiency costs across agent counts, adversary ratios, and plan numbers. Inefficiency remains low in most configurations but begins to rise when adversarial presence exceeds 70%. With fewer plans (2–4) and smaller agent populations, inefficiency increases gradually, whereas larger populations (60–100 agents) and broader plan spaces exhibit sharper increases at 100% adversaries. Expanding the number of plans (6–10) significantly enhances resilience, maintaining low inefficiency even at moderate adversarial scales (10%–80%). Peak inefficiency occurs in systems with larger populations (70–100) at 100% adversaries, reflecting reduced tolerance to adversarial influence.

Fig. 5(b) presents normalized discomfort costs under the same conditions. Discomfort increases with increasing number of plans. Systems with two plans show a gradual decline in discomfort as adversary ratios increase, while those with more plans experience sharper reductions. Discomfort variation expands with plan complexity, indicating greater sensitivity. The number of agents has minimal impact, though the 10-agent setting occasionally shows marginally lower discomfort than larger systems.

5.5. Summary of findings and discussion

The key findings of this work can be summarized as follows:

1. The interplay of adversarial scale and severity determines the resilience, vulnerability and collapse of distributed multi-objective optimization, which is strongly influenced by the optimization scenario.
2. Distributed multi-objective optimization can predominantly remain resilient, even for high adversarial scales or severity.
3. Adversarial attacks trigger high comfort losses by legitimate agents as collective compromises that reduce the likelihood of collapse for higher vulnerability and resilience.
4. Comfort compromises of legitimate agents for preserving system efficiency under adversarial attacks are predominantly required for high adversarial severity.
5. Pareto optimal points for adversarial severity levels and adversarial scales are mainly in the resilience trajectory. However, Pareto optimal points for high adversarial scales can expand to vulnerability and collapse trajectories.
6. High adversarial scales reduce the comfort compromises required by the legitimate agents in the Pareto optimal points for adversarial severity levels that can be tolerated.
7. High adversarial severity levels reduce the system efficiency in the Pareto optimal points for adversarial scales that can be tolerated.
8. Lower hierarchical levels with higher scales of agents within hierarchical structures of distributed multi-objective optimization are more vulnerable to adversarial attacks than top levels with lower scales of agents.
9. A top-down positioning of adversary agents within hierarchical structures of distributed multi-objective optimization is more impactful on system performance: higher vulnerability, likelihood of collapse, and lower resilience.
10. Systems with a small number of agents and low plan diversity are more susceptible to inefficiency increases under adversarial pressure, even at moderate adversary ratios.
11. Broader plan spaces and larger agent populations enhance system resilience in distributed multi-objective optimization, significantly suppressing inefficiency across adversarial configurations.

The findings of this paper can be used to develop and enhance corrective self-healing strategies [65] that are cost-effective in practice. They can also be used to design incentive mechanisms [66] that ensure agents comply to certain standards of safety in critical infrastructures such as smart grids. For instance, one challenge of fault-correction mechanism is the timely detection of adversary agents to mitigate their impact [65]. Apparently, redundancy mechanisms and rollback operations orchestrated by monitoring mechanisms are resource-intensive [65,67]. They require frequent checks that involve computations and exchange of messages and they usually rely on static thresholds or even manual operations [67]. This is where the insights of this work can find applicability: these mechanisms can adapt based on the status of the system, for instance, whether it is in the resilience, vulnerability and collapse state. Knowing apriori that an optimization process can tolerate certain adversarial scales and severity can simplify and reduce the costs of applying prevention and mitigation measures. It can also provide new insights for security policies, for instance, prioritizing the protection of agents at the top of the hierarchical optimization structure with stronger security safeguards and resources allocated for this purpose [68]. While creating these strategies does not fall into the scope of this paper, it is part of the future work to pursue.

6. Conclusion and future work

This study provides a comprehensive analysis of resilience, vulnerability, and collapse dynamics in multi-agent distributed optimization under adversarial conditions. By systematically examining adversary scale, severity, and network structure, we identify critical thresholds where systems transition from stability to failure. These findings offer actionable guidance for designing and enhancing the performance of recovery and healing strategies.

A key contribution of this work is the release of a large-scale benchmark dataset, generated from over 112 million experiments using the proposed adversarial model. This dataset supports systematic evaluation of adversarial impacts and facilitates reproducible research across domains of distributed optimization.

Although the adversarial model is designed to be general-purpose, the evaluation can be extended to other algorithms in future work. Additionally, the current experiments model adversarial behavior with static severity levels, which may not fully capture dynamic or strategic adversary actions. Furthermore, resilience has been analyzed under hierarchical network structures, leaving the behavior under alternative topologies an open area for exploration.

Future work will focus on embedding adaptive monitoring and mitigation mechanisms into real-time distributed systems. Investigating more complex network structures, dynamic adversarial strategies, and diverse application domains will further advance the development of robust, fault-tolerant optimization systems capable of maintaining performance under adversarial conditions.

CRediT authorship contribution statement

Amal Aldawsari: Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation. **Evangelos Pournaras:** Writing – review & editing, Project administration, Conceptualization, Methodology, Resources, Supervision, Funding acquisition.

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.

Acknowledgments

This project is funded by a UKRI Future Leaders Fellowship (MR-/W009560-/1): '*Digitally Assisted Collective Governance of Smart City Commons–ARTIO*'. The authors would like to thank Thomas Wellings and Chuhao Qin for their feedback on the paper, Abhinav Sharma for technical support in experimentation and Srijoni Majumdar for support with the voting dataset.

Appendix A. Cost function definitions

This appendix presents the mathematical definitions of the cost functions used in the optimization experiments. The functions are inherited from the original I-EPOS framework [53] and are instantiated to reflect the characteristics of each application domain.

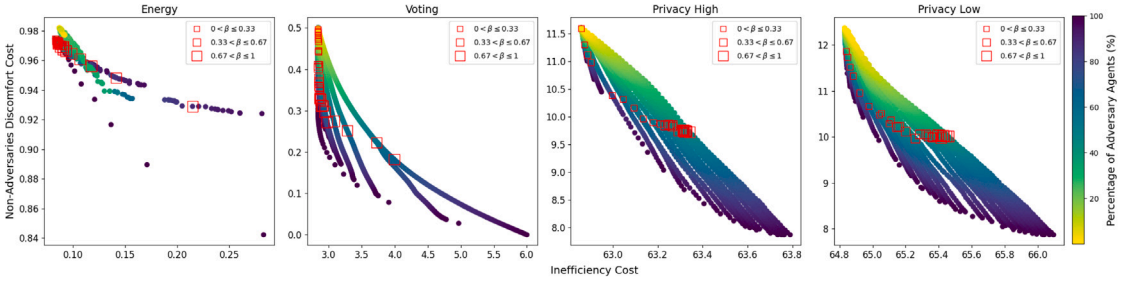
Variance

The variance cost measures the dispersion of the aggregated global response $g \in \mathbb{R}^d$ and is used in the energy and Synthetic Gaussian datasets. It is computed as:

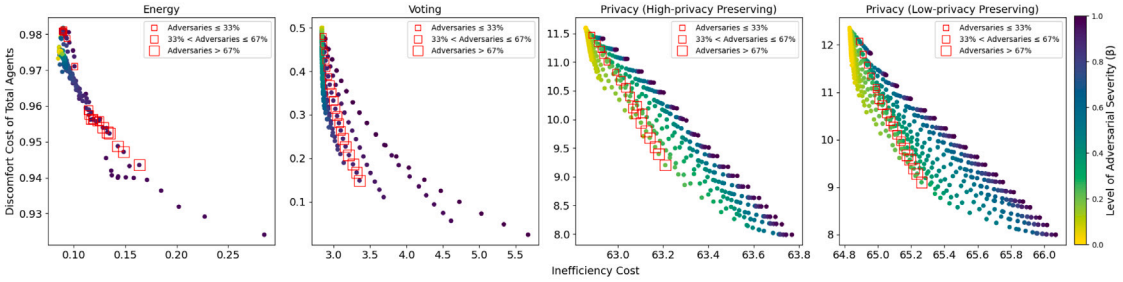
$$f_{\text{var}} = \frac{1}{d} \sum_{j=1}^{d} \left(g_j - \bar{g}\right)^2$$

where:

- g_j is the aggregated global response at dimension j,
- $\bar{g}$ is the mean of the global response across all dimensions,
- d is the dimensionality of the response.

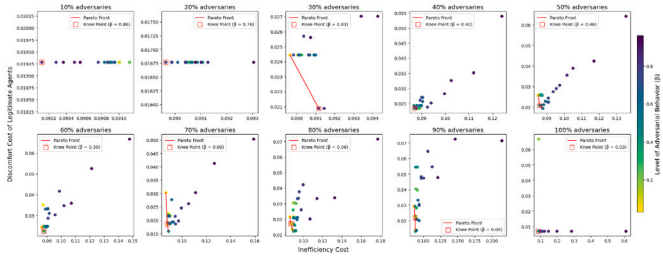


(a) Pareto front and knee points across adversarial severity.

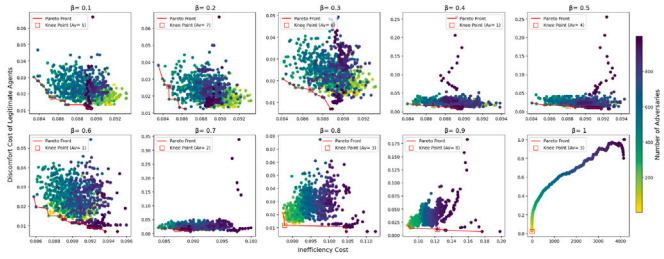


(b) Pareto front and knee points across scales of adversaries.

Fig. B.6. The pareto optimality of the energy voting, and privacy datasets.

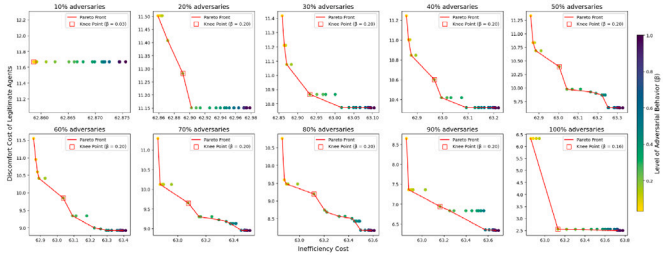


(a) Energy dataset — Pareto fronts and knee points across adversary population scales (10% to 100%).

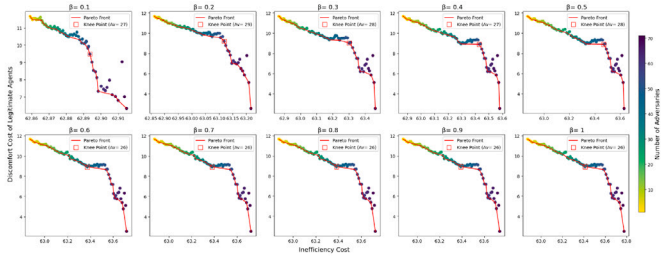


(b) Energy dataset — Pareto fronts and knee points across adversarial severity levels ($\beta = 0.1$ to 1.0).

Fig. C.7. Pareto optimality analysis for the energy dataset.



(a) Privacy dataset (high signal) — Pareto fronts and knee points across adversary population scales (10% to 100%).



(b) Privacy dataset (high signal) — Pareto fronts and knee points across adversarial severity levels ($\beta = 0.1$ to 1.0).

Fig. C.8. Pareto optimality analysis for the high privacy-preserving signal in the privacy dataset.

Residual sum of squares (RSS)

The RSS cost quantifies the squared difference between the scaled global response $g \in \mathbb{R}^d$ and a predefined system-wide target signal $T \in \mathbb{R}^d$. It is used in the voting and privacy datasets and is defined as:

$$f_{\text{RSS}} = (s(g) - s(T))^T (s(g) - s(T))$$

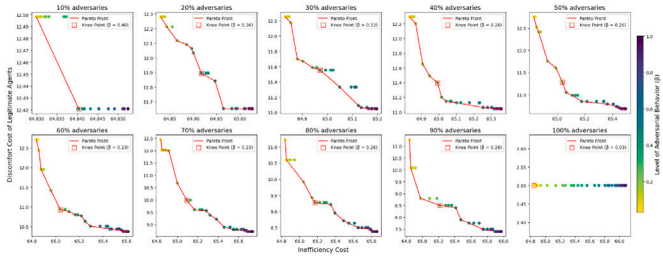
where $s(\cdot)$ denotes the scaling function applied to both vectors to improve shape alignment.

Appendix B. Pareto optimality of total agents

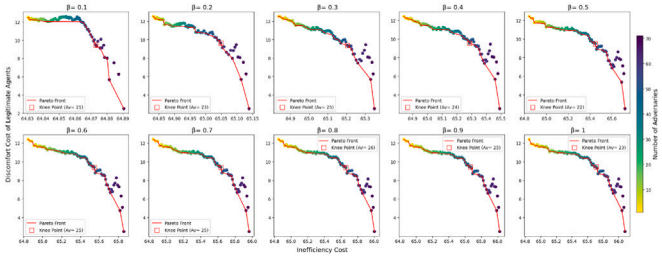
Fig. B.6 shows the trade-off between system inefficiency and total-agent discomfort, with Pareto knee points identified for both varying severity (a) and adversary scale (b).

Appendix C. Pareto optimality visualizations

This appendix presents detailed visualizations of Pareto fronts for 10 selected adversarial severity levels (β) and 10 adversary population scales across the energy, voting, and privacy dataset as shown in

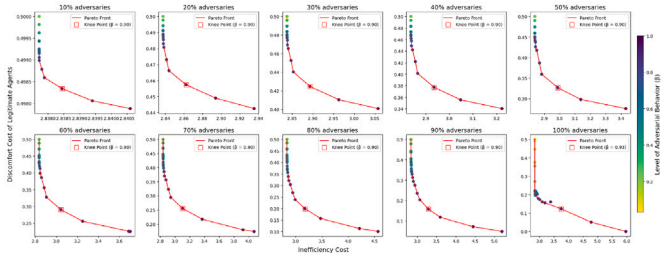


(a) Privacy dataset (low signal) — Pareto fronts and knee points across adversary population scales (10% to 100%).

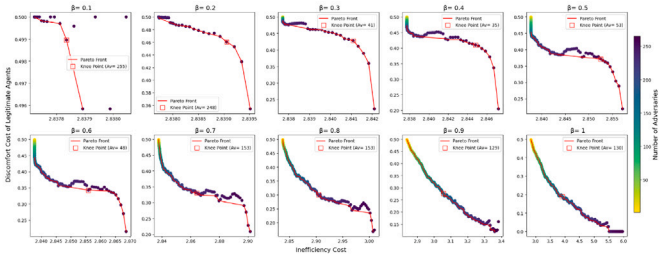


(b) Privacy dataset (low signal) — Pareto fronts and knee points across adversarial severity levels (β = 0.1 to 1.0).

Fig. C.9. Pareto optimality analysis for the low privacy-preserving signal in the privacy dataset.



(a) Voting dataset — Pareto fronts and knee points across adversary population scales (10% to 100%).



(b) Voting dataset — Pareto fronts and knee points across adversarial severity levels (β = 0.1 to 1.0).

Fig. C.10. Pareto optimality analysis for the voting dataset.

Figs. C.7–C.10. Each subfigure illustrates the trade-off between inefficiency cost and the discomfort cost of legitimate agents. Red lines indicate the non-dominated Pareto fronts, while red boxes mark the knee points, identified using the Minimum Manhattan Distance (MMD) method. These visualizations complement the analysis in Section 5, offering deeper insights into system behavior under varying adversarial conditions..

Data availability

Data will be available due course.

References

[1] Y. Perwej, K. Haq, F. Parwej, M. Mumdouh, M. Hassan, The Internet of Things (IoT) and its Application Domains, Int. J. Comput. Appl. 975 (8887) (2019) 182.

[2] M.S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, M.H. Rehmani, Applications of blockchains in the Internet of Things: A comprehensive survey, IEEE Commun. Surv. & Tutorials 21 (2) (2018) 1676–1717.

[3] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, K.H. Johansson, A survey of distributed optimization, Annu. Rev. Control. 47 (2019) 278–305.

[4] A. Nedic, A. Ozdaglar, Distributed subgradient methods for multi-agent optimization, IEEE Trans. Autom. Control 54 (1) (2009) 48–61.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends textregistered Mach. Learn. 3 (1) (2011) 1–122.

[6] G. Lan, Y. Zhou, Asynchronous decentralized accelerated stochastic gradient descent, IEEE J. Sel. Areas Inf. Theory 2 (2) (2021) 802–811, http://dx.doi.org/10.1109/JSAIT.2021.3080256.

[7] G. Lan, Y. Ouyang, Y. Zhou, Graph topology invariant gradient and sampling complexity for decentralized and stochastic optimization, SIAM J. Optim. 33 (3) (2023) 1647–1675, http://dx.doi.org/10.1137/20M138956X.

[8] N. Gupta, N.H. Vaidya, Fault-tolerance in distributed optimization: The case of redundancy, in: Proceedings of the 39th Symposium on Principles of Distributed Computing, 2020, pp. 365–374.

[9] N. Ravi, A. Scaglione, A. Nedić, A case of distributed optimization in adversarial environment, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2019, pp. 5252–5256.

[10] S. Sundaram, B. Gharesifard, Distributed optimization under adversarial nodes, IEEE Trans. Autom. Control 64 (3) (2018) 1063–1076.

[11] A.-Y. Lu, G.-H. Yang, Distributed secure state estimation in the presence of malicious agents, IEEE Trans. Autom. Control 66 (6) (2020) 2875–2882.

[12] A. Rajabi, R.B. Bobba, Resilience against data manipulation in distributed synchrophasor-based mode estimation, IEEE Trans. Smart Grid 12 (4) (2021) 3538–3547.

[13] F. Fanitabasi, A review of adversarial behaviour in distributed multi-agent optimisation, in: 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion, UCC Companion, IEEE, 2018, pp. 53–58.

[14] E. Pournaras, J. Espejo-Uribe, Self-repairable smart grids via online coordination of smart transformers, IEEE Trans. Ind. Informatics 13 (4) (2016) 1783–1793.

[15] A. González-Briones, F. De La Prieta, M.S. Mohamad, S. Omatu, J.M. Corchado, Multi-agent systems applications in energy optimization problems: A state-of-the-art review, Energies 11 (8) (2018) 1928.

[16] L. Hodgkinson, M. Mahoney, Multiplicative noise and heavy tails in stochastic optimization, in: International Conference on Machine Learning, PMLR, 2021, pp. 4262–4274.

[17] J.C. Spall, Stochastic optimization, Handb. Comput. Stat.: Concepts Methods (2012) 173–201.

[18] P.A. Hancock, Avoiding adverse autonomous agent actions, Human– Comput. Interact. 37 (3) (2022) 211–236.

[19] L. Zan, X. Zhu, Z.-L. Hu, Adversarial attacks on cooperative multi-agent deep reinforcement learning: a dynamic group-based adversarial example transferability method, Complex & Intell. Syst. 9 (6) (2023) 7439–7450.

[20] J. Nikolić, N. Jubatyrov, E. Pournaras, Self-healing dilemmas in distributed systems: Fault correction vs. fault tolerance, IEEE Trans. Netw. Serv. Manag. 18 (3) (2021) 2728–2741, http://dx.doi.org/10.1109/TNSM.2021.3092939.

[21] H. Yazdani, M. Baneshi, M. Yaghoubi, Techno-economic and environmental design of hybrid energy systems using multi-objective optimization and multi-criteria decision making methods, Energy Convers. Manage. 282 (2023) 116873.

[22] Y. Qiao, F. Hu, W. Xiong, Z. Guo, X. Zhou, Y. Li, Multi-objective optimization of integrated energy system considering installation configuration, Energy 263 (2023) 125785.

[23] N. Patari, V. Venkataramanan, A. Srivastava, D.K. Molzahn, N. Li, A. Annaswamy, Distributed optimization in distribution systems: Use cases, limitations, and research needs, IEEE Trans. Power Syst. 37 (5) (2021) 3469–3481.

[24] W. Fu, Q. Ma, J. Qin, Y. Kang, Resilient consensus-based distributed optimization under deception attacks, Internat. J. Robust Nonlinear Control 31 (6) (2021) 1803–1816.

[25] S. Zhang, Z.-W. Liu, G. Wen, Y.-W. Wang, Accelerated distributed optimization algorithm with malicious nodes, IEEE Trans. Netw. Sci. Eng. (2023).

[26] K. Kuwarancharoen, L. Xin, S. Sundaram, Byzantine-resilient distributed optimization of multi-dimensional functions, in: 2020 American Control Conference, ACC, IEEE, 2020, pp. 4399–4404.

[27] L. Su, N.H. Vaidya, Byzantine-resilient multiagent optimization, IEEE Trans. Autom. Control 66 (5) (2020) 2227–2233.

[28] J. Lin, K. Dzeparoska, S.Q. Zhang, A. Leon-Garcia, N. Papernot, On the robustness of cooperative multi-agent reinforcement learning, in: 2020 IEEE Security and Privacy Workshops, SPW, IEEE, 2020, pp. 62–68.
[29] M. Figura, K.C. Kosaraju, V. Gupta, Adversarial attacks in consensus-based multi-agent reinforcement learning, in: 2021 American Control Conference, ACC, IEEE, 2021, pp. 3050–3055.
[30] Y. Zheng, Z. Yan, K. Chen, J. Sun, Y. Xu, Y. Liu, Vulnerability assessment of deep reinforcement learning models for power system topology optimization, IEEE Trans. Smart Grid 12 (4) (2021) 3613–3623.
[31] H. Ishii, Y. Wang, S. Feng, An overview on multi-agent consensus under adversarial attacks, Annu. Rev. Control. 53 (2022) 252–272.
[32] M. Yemini, A. Nedić, S. Gil, A.J. Goldsmith, Resilience to malicious activity in distributed optimization for cyberphysical systems, in: 2022 IEEE 61st Conference on Decision and Control, CDC, 2022, pp. 4185–4192, http://dx.doi.org/10.1109/CDC51059.2022.9992416.
[33] K. Du, Q. Ma, Y. Kang, S. Wang, A distributed optimization algorithm over Markov switching topology under adversarial attack, J. Franklin Inst. 360 (16) (2023) 12770–12784.
[34] C. Zhao, J. He, Q.-G. Wang, Resilient distributed optimization algorithm against adversarial attacks, IEEE Trans. Autom. Control 65 (10) (2019) 4308–4315.
[35] C.A. Uribe, H.-T. Wai, M. Alizadeh, Resilient distributed optimization algorithms for resource allocation, in: 2019 IEEE 58th Conference on Decision and Control, CDC, IEEE, 2019, pp. 8341–8346.
[36] B. Turan, C.A. Uribe, H.-T. Wai, M. Alizadeh, Resilient primal–dual optimization algorithms for distributed resource allocation, IEEE Trans. Control. Netw. Syst. 8 (1) (2020) 282–294.
[37] R. Gentz, S.X. Wu, H.-T. Wai, A. Scaglione, A. Leshem, Data injection attacks in randomized gossiping, IEEE Trans. Signal Inf. Process. over Networks 2 (4) (2016) 523–538.
[38] T. Ding, S. Zhu, J. He, C. Chen, X. Guan, Consensus-based distributed optimization in multi-agent systems: Convergence and differential privacy, in: 2018 IEEE Conference on Decision and Control, CDC, IEEE, 2018, pp. 3409–3414.
[39] T. Ding, S. Zhu, J. He, C. Chen, X. Guan, Differentially private distributed optimization via state and direction perturbation in multiagent systems, IEEE Trans. Autom. Control 67 (2) (2021) 722–737.
[40] A. Guesmi, M.A. Hanif, B. Ouni, M. Shafique, Physical adversarial attacks for camera-based smart systems: Current trends, categorization, applications, research challenges, and future outlook, IEEE Access (2023).
[41] H. Ali, D. Chen, M. Harrington, N. Salazar, M. Al Ameedi, A.F. Khan, A.R. Butt, J.-H. Cho, A survey on attacks and their countermeasures in deep learning: Applications in deep neural networks, federated, transfer, and deep reinforcement learning, IEEE Access 11 (2023) 120095–120130.
[42] K. Fettah, T. Guia, A. Salhi, A. Betka, A.S. Saidi, M. Teguar, E. Ali, M. Bajaj, S.A.D. Mohammadi, S.S. Ghoneim, A pareto strategy based on multi-objective optimal integration of distributed generation and compensation devices regarding weather and load fluctuations, Sci. Rep. 14 (1) (2024) 10423.
[43] C. Zhang, H. Liu, S. Pei, M. Zhao, H. Zhou, Multi-objective operational optimization toward improved resilience in water distribution systems, AQUA— Water Infrastruct. Ecosyst. Soc. 71 (5) (2022) 593–607.
[44] S.P. Boindala, A. Ostfeld, Robust multi-objective design optimization of water distribution system under uncertainty, Water 14 (14) (2022) 2199.
[45] T. Asikis, E. Pournaras, Optimization of privacy-utility trade-offs under informational self-determination, Future Gener. Comput. Syst. 109 (2020) 488–499.
[46] C. Zhang, Y. Wang, Enabling privacy-preservation in decentralized optimization, IEEE Trans. Control. Netw. Syst. 6 (2) (2018) 679–689.
[47] Y. Lu, M. Zhu, Privacy preserving distributed optimization using homomorphic encryption, Automatica 96 (2018) 314–325.
[48] S. Mao, Y. Tang, Z. Dong, K. Meng, Z.Y. Dong, F. Qian, A privacy preserving distributed optimization algorithm for economic dispatch over time-varying directed networks, IEEE Trans. Ind. Informatics 17 (3) (2020) 1689–1701.
[49] X. Chen, L. Huang, L. He, S. Dey, L. Shi, A differentially private method for distributed optimization in directed networks via state decomposition, IEEE Trans. Control. Netw. Syst. (2023).
[50] C. Hinrichs, S. Lehnhoff, M. Sonnenschein, COHDA: A combinatorial optimization heuristic for distributed agents, in: Agents and Artificial Intelligence: 5th International Conference, ICAART 2013, Barcelona, Spain, February 15-18, 2013. Revised Selected Papers 5, Springer, 2014, pp. 23–39.
[51] C. Hinrichs, M. Sonnenschein, A distributed combinatorial optimisation heuristic for the scheduling of energy resources represented by self-interested agents, Int. J. Bio- Inspired Comput. 10 (2) (2017) 69–78.
[52] E. Pournaras, M. Yao, D. Helbing, Self-regulating supply–demand systems, Future Gener. Comput. Syst. 76 (2017) 73–91.
[53] E. Pournaras, P. Pilgerstorfer, T. Asikis, Decentralized collective learning for self-managed sharing economies, ACM Trans. Auton. Adapt. Syst. (TAAS) 13 (2) (2018) 1–33.
[54] E. Pournaras, Collective learning: A 10-year odyssey to human-centered distributed intelligence, in: 2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems, ACSOS, IEEE, 2020, pp. 205–214.
[55] E. Pournaras, S. Yadhunathan, A. Diaconescu, Holarchic structures for decentralized deep learning: a performance analysis, Clust. Comput. 23 (2020) 219–240.
[56] F. Fanitabasi, E. Pournaras, Appliance-level flexible scheduling for socio-technical smart grid optimization, IEEE Access 8 (2020) 119880–119898.
[57] E. Pournaras, Multi-Level Reconfigurable Self-Organization in Overlay Services (Ph.D. thesis), Delft University of Technology, School of Technology Policy and Management, Department of Multi-Actor Systems, 2013, http://dx.doi.org/10.12681/eadd/30107.
[58] E. Pournaras, M.C. Ballandies, S. Bennati, C.-f. Chen, Collective privacy recovery: Data-sharing coordination via decentralized artificial intelligence, PNAS Nexus 3 (2) (2024) pgae029.
[59] S. Majumdar, E. Pournaras, Score Voting Plans - 256 Voters in UK Labor 2010 Elections, figshare, 2024, http://dx.doi.org/10.6084/m9.figshare.27925494.v1.
[60] C. Navarrete, M. Macedo, R. Colley, J. Zhang, N. Ferrada, M.E. Mello, R. Lira, C. Bastos-Filho, U. Grandi, J. Lang, et al., Understanding political divisiveness using online participation data from the 2022 French and Brazilian presidential elections, Nat. Hum. Behav. 8 (1) (2024) 137–148.
[61] Y. Sun, S. Li, A knee-oriented many-objective differential evolution with bi-strategy and manhattan distance-domination range, Swarm Evol. Comput. 89 (2024) 101637.
[62] W. Li, G. Zhang, T. Zhang, S. Huang, Knee point-guided multiobjective optimization algorithm for microgrid dynamic energy management, Complexity 2020 (1) (2020) 8877008.
[63] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66, http://dx.doi.org/10.1109/TSMC.1979.4310076.
[64] M.H. Merzban, M. Elbayoumi, Efficient solution of Otsu multilevel image thresholding: A comparative study, Expert Syst. Appl. 116 (2019) 299–309.
[65] R.R. Nejad, W. Sun, Enhancing active distribution systems resilience by fully distributed self-healing strategy, IEEE Trans. Smart Grid 13 (2) (2021) 1023–1034.
[66] S. Bhattacharya, R. Chengoden, G. Srivastava, M. Alazab, A.R. Javed, N. Victor, P.K.R. Maddikunta, T.R. Gadekallu, Incentive mechanisms for smart grid: State of the art, challenges, open issues, future directions, Big Data Cogn. Comput. 6 (2) (2022) 47.
[67] H. Liang, X. Yin, Self-healing control: Review, framework, and prospect, IEEE Access 11 (2023) 79495–79512.
[68] S.Y. Enoch, J. Mendonça, J.B. Hong, M. Ge, D.S. Kim, An integrated security hardening optimization for dynamic networks using security and availability modeling with multi-objective algorithm, Comput. Netw. 208 (2022) 108864.

Amal Aldawsari is a Ph.D. researcher in the School of Computer Science at the University of Leeds, United Kingdom. She received her B.Sc. degree with first-class honors in Computer Science from the College of Computer Science and Engineering, University of Hail (UOH), Saudi Arabia, in 2014, and her M.Sc. degree in Computer Science from King Saud University (KSU), Riyadh, Saudi Arabia, in 2020. From 2015 to 2017, she served as a teaching assistant at UOH, and later worked as a Lecturer from 2020 to 2022. Her current research focuses on distributed optimization and multi-agent systems, aiming to improve efficiency, resilience, and decision-making in large-scale intelligent systems.

Dr. Evangelos Pournaras is Professor of Trustworthy Distributed Intelligence in the School of Computer Science at University of Leeds. He is also a UKRI Future Leaders Fellow (£1.4M), a Research Associate at the UCL Center of Blockchain Technologies and has also been an Alan Turing Fellow. Evangelos has more than 5 years of research experience at ETH Zurich after having completed his Ph.D. studies at Delft University of Technology. Evangelos has also been a visiting researcher at EPFL and has industry experience at IBM T.J. Watson Research Center. Evangelos has won the Augmented Democracy Prize, the 1st prize at ETH Policy Challenge as well as 5 paper awards and honors, including the listing of two of his project within UNESCO IRCAI Global Top-100 as 'outstanding' and 'promising'. He has published more than 100 peer-reviewed papers in high impact journals and conferences. Evangelos has extensive leadership experience and raised funding for national and EU projects such as H2OforAll, ASSET and SoBigData.