Reliability Engineering and System Safety 274 (2026) 112444

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

Resilience evaluation and recovery strategy optimization for unmanned system-of-systems considering relay communication

Xu Wang [a], Hengte Du [b], Yuheng Dang [a], Xing Pan [a,*]

[a] School of Reliability and Systems Engineering, Beihang University, Beijing, China
[b] Department of Industrial Engineering, Tsinghua University, Beijing, China

ARTICLE INFO

Keywords:
Resilience
Unmanned system-of-systems
Multi-agent
Bi-objective optimization algorithm

ABSTRACT

Unmanned System-of-Systems (USoS) equipped with a relay communication system exhibit pronounced structural coupling and heightened susceptibility to cascading failures. Existing evaluation approaches inadequately capture the functional role of relay entities, and prevailing recovery strategies show limited adaptability to USoS imposed by relay-enabled architectures. To address these limitations, this study develops an integrated framework for resilience evaluation and recovery strategy optimization. A Task-Driven Multilayer Network (TDMN) model is constructed to quantify relay-induced communication gains and task-chain dependencies. Building on this foundation, a bi-objective Moth–Flame Optimization algorithm BMRMFO is developed to simultaneously minimize cumulative performance loss resilience and total recovery time cost. The algorithm integrates initialization based on maintain rules and local reinforcement mechanisms to enhance convergence stability within discrete recovery spaces. Multi-scale USoS experiments show that TDMN can effectively quantify relay-induced performance gains, thereby highlighting the necessity of incorporating task-level mechanisms into system evaluation. Building on this evaluation foundation, the proposed framework consistently outperforms nine baseline algorithms. In particular, it achieves 27.62% and 22.61% improvements on two objectives, respectively. Statistical significance tests and relay-priority analyses further confirm its suitability for relay-enabled USoS, where relay-driven structural coupling and task-chain dependencies shape mission-level performance and recovery decisions.

1. Introduction

With the rapid advancement of unmanned and communication technologies, Unmanned System-of-Systems (USoS) have evolved toward highly distributed, intelligent, and emergent operational patterns [1]. Owing to their lightweight design, miniaturization, customization, and cost efficiency, unmanned platforms have become key enablers of next-generation wireless communication systems and, when deployed as mobile relay entities, can substantially enhance communication performance and extend operational reach [2,3]. Compared with traditional ground-based or satellite relays, these mobile relay systems offer broader coverage and greater adaptability, and have been widely applied in reconnaissance, terrain mapping, and disaster-response missions [4–7]. However, recent studies indicate that the introduction of relay entities fundamentally reshapes network connectivity, multi-hop routing structures, and load distribution, thereby influencing system vulnerability and cascading-failure dynamics in IoT, IIoT, and multi-hop wireless networks [8–12]. These relay-induced structural and dynamical effects present significant challenges for sustaining reliable USoS operations.

In resilience engineering, performance evaluation and recovery optimization jointly determine a system's capacity to withstand, absorb, and recover from disruptions [13]. Although these issues have been widely examined in critical infrastructures [14,15], supply chains [16], and unmanned systems [17–19], they present heightened complexity in relay-enabled USoS. Existing performance evaluation models fail to adequately capture the communication gains introduced by the Relay Communication System, leading to inaccurate capability assessment. Furthermore, relay-enabled architectures increase structural complexity and amplify reliability risks, making mission failures more likely under external disturbances or internal faults and thereby heightening the need for timely and effective recovery strategies to reconfigure the system architecture and preserve essential functions [20,21]. Therefore, an integrated framework that unifies USoS performance evaluation and

* Corresponding author.
E-mail address: panxing@buaa.edu.cn (X. Pan).

https://doi.org/10.1016/j.ress.2026.112444
Received 10 September 2025; Received in revised form 15 February 2026; Accepted 17 February 2026
Available online 22 February 2026

recovery optimization is urgently needed to address the substantial risks that relay-induced structural and dynamical changes pose to system reliability across diverse scenarios.

To accurately evaluate the resilience of USoS, a performance evaluation framework is required that captures how system functionality evolves under disruptions. Traditional indicators for unmanned systems, such as operation loops derived from the Observe–Orient–Decide–Act (OODA) framework, reflect task concurrency and functional diversity within a given time window [22]. However, excessively long loops diminish operational efficiency, and simple loop counts provide limited insight into how collaborative behaviors or information constraints shape system performance. Although recent studies have attempted to eliminate redundant loops and improve evaluation precision [23], these efforts remain confined to isolated functional capabilities and thus offer limited explanatory power at the mission level. The critical role of relay entities further heightens USoS vulnerability, underscoring the need to understand how relay-induced communication constraints and structural dependencies influence performance and resilience. Existing research, however, primarily examines cascading failures and survivability at the communication or network layer, leaving their implications for mission-level performance evaluation and recovery decision making in relay-enabled USoS insufficiently explored [24]. With continued progress in system research, a task completion capability–centered evaluation paradigm has emerged. Existing mission-level performance evaluation methods can be broadly classified into two categories. The first is task semantic-based evaluation, which selects indicators according to specific task concepts, such as reconnaissance coverage or loitering time. The second is operation loop-based evaluation, which estimates effectiveness by measuring task baselines and completion probabilities. Notably, tasks represent a higher-level abstraction of system capabilities [25] and are structurally interconnected through task chains [26]. The introduction of task chains reveals that localized improvements in individual capabilities do not necessarily enhance overall mission performance; substantial gains in task completion capability can only be achieved through coordinated evolution of multiple capabilities within a task chain [27,28]. In this context, it is necessary to develop modeling and evaluation approaches from a task chain perspective to support the characterization and analysis of coordinated capability evolution in complex USoS scenarios.

Existing resilience optimization approaches can be broadly grouped into importance-driven, reinforcement learning, and heuristic optimization methods. Importance-based methods feature low computational complexity and high interpretability, as they generate recovery strategies according to the contribution of individual entities to system resilience [15,29]. However, their applicability is generally limited to relatively simple scenarios and does not extend well to complex, multi-objective decision environments. Reinforcement learning approaches enable adaptive and iterative policy optimization without requiring complete knowledge of task scenarios [17,30]; yet, they typically demand long training periods, exhibit sensitivity to scenario variations, and suffer from limited generalizability across heterogeneous operational conditions. In contrast, heuristic optimization algorithms—valued for their simple implementation and rapid convergence—have been widely applied in USoS recovery strategy design [2,31], offering greater flexibility and interpretability [32–34]. However, most existing applications directly employ generic metaheuristics without explicitly modeling the relay-induced structural coupling, multi-hop communication constraints and heterogeneous recovery measures inherent to USoS recovery scenarios. Consequently, the search process often fails to leverage problem-specific structural features and may yield recovery strategies that are suboptimal from a resilience perspective. Moreover, contemporary system-level design efforts tend to emphasize performance enhancement, while the interplay between system architecture and resilience remains insufficiently examined [35,36].

To address the challenges mentioned above, this paper develops a modeling, evaluation, and optimization method aimed at enhancing the resilience of the USoS. The main contributions are as follows:

(i) . A Task-driven Multilayer Network (TDMN) is developed to evaluate the performance of USoS with relay communication. By explicitly modeling relay-communication relationships and introducing a new operation-loop analysis method, the model quantifies the performance gains enabled by relays. In addition, TDMN captures both structural dependencies and core mission demands, establishing a task-centric performance evaluation framework through the construction of task chains.

(ii) . To address the recovery optimization problem in USoS, this study proposes a bi-objective recovery optimization algorithm that integrates four heterogeneous recovery measures—Functional Switching, Resource Allocation, Redundant Activation, and On-site Maintenance—to jointly minimize cumulative performance loss resilience and total recovery time cost. Built upon the Moth–Flame Optimization (MFO) framework, the proposed method develops a discrete solution representation for MFO to match the recovery optimization model established in this work, thereby forming BMRMFO. On this basis, a maintenance rule-guided initialization scheme and a local reinforcement mechanism are further incorporated to improve feasibility and search efficiency in the discrete decision space.

(iii) . Extensive multi-scale USoS case studies spanning different disruption intensities and support-resource settings evaluate the proposed framework. The evaluation model is validated by demonstrating relay-induced structural evolution and mission-level coordination patterns, highlighting the need for task-chain–aware assessment. Building on this, BMRMFO is benchmarked against four bi-objective algorithms and six baselines, with added analyses on whether relay-related elements are naturally prioritized; the results consistently reveal relay-driven trends and emergent relay-neighborhood prioritization, demonstrating superior overall recovery performance.

The remainder of this paper is organized as follows. Section 2 introduces the TDMN model and the associated system performance evaluation approach. Section 3 presents the USoS recovery scenario and the proposed bi-objective recovery strategy optimization algorithm BMRMFO. Section 4 presents case studies and comparative experiments validating TDMN and BMRMFO. Finally, Section 5 concludes the paper and outlines future research directions.

2. Modeling and evaluation of USoS considering relay communication

2.1. TDMN Model

2.1.1. Modeling framework

The USoS is fundamentally designed to accomplish prescribed core missions; therefore, its performance must be assessed through mission fulfillment. Drawing inspiration from multilayer network theory [2,3], this study develops TDMN that quantitatively evaluates communication-enabled capability gains while reflecting the mission-oriented and task-driven characteristics of the USoS. TDMN is defined as a three-layer interconnected network $\mathscr{N} = (\mathscr{S}, \mathscr{C}, \mathscr{M})$. The Structure Layer $\mathscr{S}$, modeled as an unweighted undirected network, characterizes the direct communication links and functional interactions among physical entities. The Capability Layer $\mathscr{C}$, modeled as a weighted directed network, incorporates relay communication mechanisms and the OODA operational framework, where edge weights represent the required number of communication hops between entities. The Mission Layer $\mathscr{M}$, modeled as an unweighted directed network, represents the tasks and task chains that the USoS must perform to accomplish its core missions, represents the mission demands obtained through mission

decomposition. By separating structural configurations, communication topology, capability interactions, and mission demand, the TDMN provides an accurate and systematic framework for USoS performance evaluation.

Fig. 1 illustrates the overall structure of the USoS multilayer network. In this framework, the Mission Layer derives mission demands by decomposing core missions into task chains, thereby providing the criteria for assessing mission fulfillment. The Structure Layer represents the real operational environment by describing the number, functional roles, spatial distribution, and interaction relationships of friendly and target entities. Based on the performance attributes of physical entities and the identified mission demands, the Capability Layer applies information-flow rules to filter and map feasible links, abstracting individual entity functions into operation loops that support specific tasks and establishing the correspondence between mission demands and physical entities.

The TDMN provides a unified and rigorous framework for evaluating USoS performance. By explicitly modeling relay mechanisms, it quantifies relay-enabled performance gains and assesses their influence on mission fulfillment. Furthermore, the Mission Layer specifies mission demands and links them to the structural entities, enabling performance assessment results to inform the selection and prioritization of recovery actions and to guide the design of more resilient USoS architectures.

2.1.2. *Modeling process*

(1) Structure Layer Modeling

USoS performance evaluation relies on comprehensive information acquisition; therefore, Structure Layer modeling must be conducted first to identify entity attributes and interaction relationships. According to the OODA framework, entities can be categorized into four types: sensors S, which perform early intelligence, surveillance, and reconnaissance functions; deciders D, which undertake command and control roles; influencers I, directly interact with and alter the physical environment; targets T, representing mission-determined interactive objects external to the USoS.

The Structure Layer network is defined as $\mathscr{S} = (V_{\mathscr{S}}, E_{\mathscr{S}})$, where the node set $V^{\mathscr{S}} = V^S \cup V^D \cup V^I \cup V^T$ represents the entities, V^S, V^D, V^I and V^T denote the sets of sensor entities, decider entities, influencer entities, and target entities, respectively. The edge set $E^{\mathscr{S}} = E^C \cup E^F$ defines interactions among entities, primarily consists of functional interaction edges E^F and communication interaction edges E^C. Fig. 2 depicts the interaction relationships between entities.

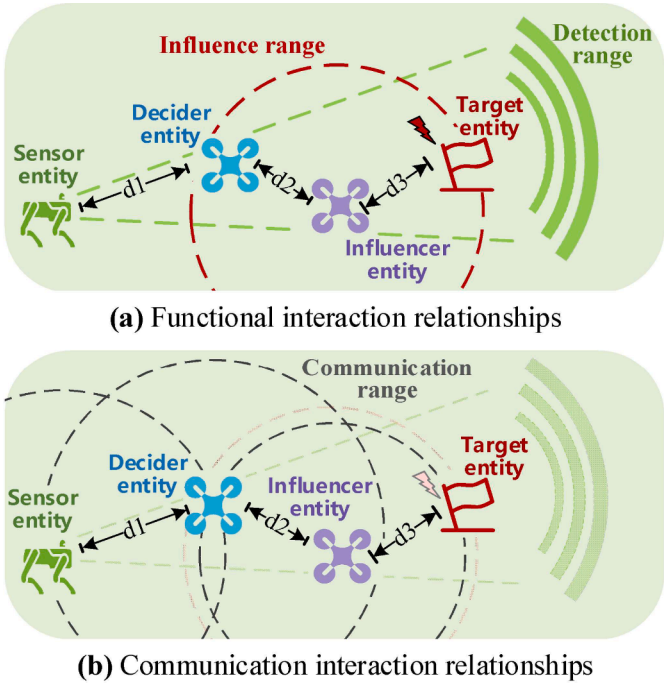


Fig. 2. Entity interaction relationships.

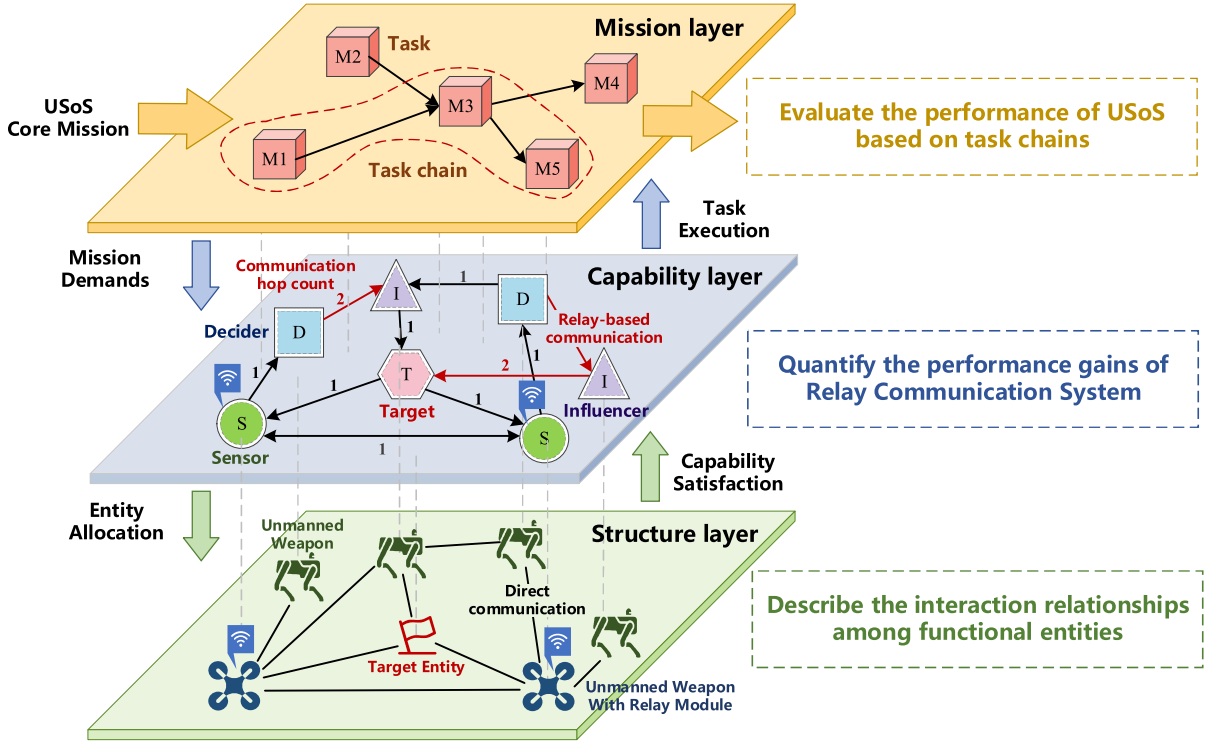


Fig. 1. Task-driven multilayer network.

In Fig. 2 (a), functional interaction relationships are established when a target entity falls within the detection range of sensor entities or the influence range of influencer entities, indicating functional connectivity between the entities. For example, when the distance d3 between the influencer entity and the target entity is less than the influence range, a functional interaction relationship is formed. As illustrated in Fig. 2 (b), the establishment of communication interaction relationships is governed by factors such as distance, communication range, and the importance of unmanned entities, thus leading to a preferential connection tendency toward critical or geographically proximate nodes. Specifically, if the unmanned entity v_{i2} lies within the communication range of the entity v_{i1}, the probability $\prod(\text{degree}_{i1}|i2)$ of establishing a communication link between v_{i1} and v_{i2} can be calculated using Eq. (1) [37]:

$$\prod(\text{degree}_{i1}|i2) = \beta\frac{\text{degree}_{i1}}{\sum_{n_e=1}^{N_t} k_{n_e}} + (1-\beta)\frac{\text{Fun}(d_{i2,i1})}{\sum_{n_e=1}^{N_t}\text{Fun}(d_{i2,n_e})} \tag{1}$$

where, $\beta \in (0,1]$ denotes the weighting factor between node degree and distance. When $\beta = 1$, the probability depends solely on the degree of each unmanned entity. When $0 < \beta < 1$, the model incorporates both the node degree and a distance-based function. The parameter β thus determines whether the model prioritizes connections with critical nodes or establishes links based on communication relationships. In particular, $degree_{i1}$ represents the degree of entity $i1$. N_t is the number of unmanned entities at time t, and $d_{i2,i1}$ denotes the distance between unmanned entity $i1$ and $i2$. The function $Fun(d_{i2,i1})$ serves as a distance normalization function, which adjusts the pairwise distances according to communication range constraints.

(2) Capability Layer Modeling

The Capability Layer abstracts the discrete physical components within the USoS and integrates originally independent functional entities, thereby establishing a formal correspondence between individual entity functions and USoS-level capabilities [38]. At this level, the information-flows in this layer represent various functional interactions, including reconnaissance sharing $\{S \leftrightarrow S\}$, target detection $\{S \rightarrow D\}$, decision sharing $\{D \leftrightarrow D\}$, command delivery $\{D \rightarrow I\}$, target engagement $\{I \rightarrow T\}$, and performance feedback $\{T \rightarrow S\}$ [2]. These functional interactions in the Capability Layer do not exist in isolation; instead, they are organized according to specific causal logic to form a closed-loop chain of operational execution, namely the meta-operation loop.

Definition 2.1. **(*Meta-operation loop*).** A meta-operation loop $c = (\breve{E}, \Phi)$ is a directed cyclic path which represents a series of directed edges among node types: $U_1 \xrightarrow{\varphi_1} U_2 \xrightarrow{\varphi_2} \ldots \xrightarrow{\varphi_r} U_{r+1} \xrightarrow{\varphi_{r+1}} U_1$, where node types $U_1, U_2, \ldots, U_{r+1} \in \breve{E}$, and link types $\varphi_1, \varphi_2, \ldots, \varphi_r \in \Phi$.

Common meta-operation loops are illustrated in Fig. 3. Meta-operation loop (a) depicts a basic operation loop $\{S \rightarrow D \rightarrow I \rightarrow T \rightarrow S\}$: a sensor detects a target and transmits relevant information to a decider, which issues a command to an influencer, who then exerts disruptive effects on the target. Based on this structure, Meta-operation loop (b) $\{S \rightarrow D \rightarrow D \rightarrow I \rightarrow T \rightarrow S\}$ incorporates collaborative decision-making, meta-operation loop (c) $\{S \rightarrow S \rightarrow D \rightarrow I \rightarrow T \rightarrow S\}$ adds cooperative reconnaissance, and meta-operation loop (d) $\{S \rightarrow S \rightarrow D \rightarrow D \rightarrow I \rightarrow T \rightarrow S\}$ combines both enhancements, resulting in four fundamental meta-operation loops.

In relay-enble USoS architectures, interaction relationships become highly complex; beyond the functional entities, the communication coupling introduced by relay entities must also be considered. Therefore, the Capability Layer can be formally defined as $\mathscr{C} = (V^{\mathscr{C}}, E^{\mathscr{C}}, W^{\mathscr{C}})$, where the node set is classified by relay configuration: $V^{\mathscr{C}} = V^{R;S,D,I} \cup V^{N;S,D,I} \cup V^{T}$. Here, $V^{R;S,D,I}$ represent the relay entities with sensing, decision-making, and influencing functions, respectively, while $V^{N;S,D,I}$ denote their non-relay counterparts. The edge set captures five categories of information-flows, denoted as: $E^{\mathscr{C}} = \{S \leftrightarrow S\} \cup \{S \rightarrow D\} \cup \{D \leftrightarrow D\} \cup \{D \rightarrow I\} \cup \{I \rightarrow T\}$. Each edge represents one of the following flows: reconnaissance sharing, target detection, decision sharing, command delivery, and strike assessment. A weight function $W^{\mathscr{C}} = \{w^{\mathscr{C}}(e^{\mathscr{C}}) | e^{\mathscr{C}} \in E^{\mathscr{C}}\}$ assigns a positive integer to each edge, representing the number of communication hops required between two entities, equivalent to the length of the shortest relay-based path. Specifically, $w^{\mathscr{C}}(e^{\mathscr{C}}) = 1$ denotes direct interaction without relays, whereas $w^{\mathscr{C}}(e^{\mathscr{C}}) > 1$ indicates communication through one or more intermediate relay entities.

Information flows in the Capability Layer originate from the interaction relationships defined in the Structure Layer and are retained only when both functional compatibility and communication feasibility are satisfied. In USoS considering relay communication, feasible information exchange may occur through direct links or through relay-supported paths composed exclusively of relay entities. To identify such paths, for each ordered entity pair (*src,tgt*) without a direct link in the Structure Layer, we search for the shortest relay path P_{relay} whose internal nodes are all in the relay set V_0. If P_{relay} exists and its hop count is no greater than max hops, we add a directed edge (*src,tgt*) to the Capability Layer and set its weight to the hop count. After processing all candidate pairs, we filter the resulting edges by the information-flow interaction rules and remove any links that violate the type constraints or exceed the hop threshold. The remaining directed weighted graph forms the finalized Capability Layer, as summarized in Algorithm 1.

Once the Capability Layer network is constructed, operation loops are identified to support performance evaluation. As noted earlier, the introduction of relay entities improves network connectivity and enhances overall USoS performance. To further demonstrate the performance gains of the Relay Communication System, we conduct comparative analyses under different relay configurations. Assuming the number of relay entities is N_R, Relay Communication System with $N_R = 0$, $N_R = 1$ and $N_R = 2$ was configured in the instance. The topology of the Capability Layer under these three instances is shown in Fig. 4.

In Fig. 4, the directed and weighted edges represent the Capability Layer network generated by Algorithm 1. Black edges denote interaction links that exist in the USoS without relay entities; red edges correspond to new links enabled by the presence of relay entities; and grey edges indicate direct communication links removed due to violations of information-flow rules. Notably, indirect communication through these

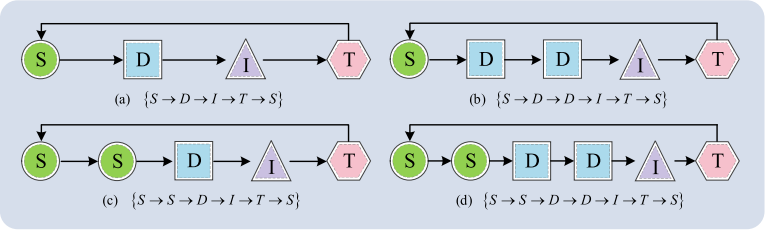


Fig. 3. Four common meta-operation loops.

Algorithm 1
Capability layer network generation.

Input: Structure Layer network $\mathscr{S}$

Relay entity set V_0
Information flow rules *interaction_rules*
Maximum allowable hop count *max_hops*
Output: Capability Layer Network $\mathscr{C}$
Add all nodes in $\mathscr{S}$ to $\mathscr{C}$
foreach *edge* $(u, v) \in \mathscr{S}$ **do**
    Add directed *edge* (u, v) and (v, u) to $\mathscr{C}$ with weight 1
end
foreach (src, tgt) in $\mathscr{S}$ where $src \neq tgt$ **do**
    Find the shortest path P_{relay} from *src* to *tgt*
        Subject to: all internal nodes of $P_{\text{relay}} \in V_0$
    if P_{relay} *exists* and *Length*$(P_{relay}) \leq max\_hops$ **then**
        Add directed *edge* (src, tgt) to $\mathscr{C}$
        Set *weight* (src, tgt) = Length (P_{relay})
    end
end
foreach *edge* (u, v) in $\mathscr{C}$ **do**
    if *type* $(v) \notin$ *interaction_rules* [*type* (u)] **then**
        Remove *edge* (u, v)
    end
end
return $\mathscr{C}$

grey links remains possible if a valid relay path exists. In this case study, the maximum allowable number of communication hops is set to two. The introduction of additional relay entities substantially increases the number of links in the Capability Layer network. In addition, a further examination of whether the proposed algorithm can quantify the performance gains introduced by the Relay Communication System will be presented in Section 4.2.1.

(3) Mission Layer Modeling

Within TDMN, the Mission Layer represents broad strategic goals that define the fundamental purpose and long-term direction of the system. In this context, tasks are discrete, well-defined elements within the mission layer that correspond to specific activities required to realize components of these strategic goals. Thus, while missions establish the overarching strategic framework, tasks constitute the concrete operational steps necessary for mission fulfillment.

Definition 2.2. **(*Meta-task chain*).** Meta-task chain is an ordered sequence of task types which is described as $\mathscr{P}$. Formally, the meta-task chain p can be expressed as Eq. (2):

$$\mathscr{P}_p = \left\langle L_1^{(p)}, L_2^{(p)}, \ldots, L_{m_p}^{(p)} \right\rangle (p = 1, 2, \ldots, n_p) \tag{2}$$

where $L^{(p)}$ denotes the task type at the one stage of the meta-task chain. m_p denotes the number of task types of meta-task chain p. n_p denotes the number of meta-task chain.

In real-world applications, tasks may involve sequential dependencies or parallel coordination, leading to diverse meta-task chains. For example, in emergency rescue scenarios, area mapping or communication network setup must often be completed before rescue operations can proceed. Based on the four types of meta-operation loops previously constructed, four corresponding task types can be defined: Basic Tasks, which are relatively straightforward and foundational; Sensing Tasks, which require accurate target observation and perception; Decision Tasks, which involve complex reasoning and multi-factor evaluation; and Sensing-Decision hybrid Tasks, which integrate perceptual and decision-making requirements.

These task types can be combined according to target characteristics to construct meta-task chains, which are executed in appropriate quantities to fulfill the core missions. Each operation loop within the Capability Layer is abstracted into a task instance in the Mission Layer, representing a specific task supported by the underlying capabilities. To illustrate the process by which task targets are decomposed into task sequences and corresponding task demands are established, Fig. 5 presents a schematic example of this task-decomposition process. For illustration, $\langle L_1, L_2, L_3, L_4 \rangle$ represents one such meta-task chain; in practice, meta-task chains can be flexibly designed to meet the specific requirements of each core mission.

Based on the mapping between operation loops and task types, the Mission Layer network is defined as $\mathscr{M} = (V^{\mathscr{M}}, E^{\mathscr{M}})$, where the node set $V^{\mathscr{M}} = V^{L_1} \cup V^{L_2} \cup V^{L_3} \cup V^{L_4}$ consists of four task types. A logical connection between two tasks is created only when they are associated with the same target entity. Therefore, multiple tasks can be organized into a meta-task chain only if their corresponding operation loops are directed toward the same target entities. Accordingly, the edge set $E^{\mathscr{M}} = \left\{ e_1^{\mathscr{M}}, e_2^{\mathscr{M}}, \ldots, e_{N^{\mathscr{M}}}^{\mathscr{M}} \right\}$, where $N^{\mathscr{M}}$ denotes the number of links in the Mission Layer, $E^{\mathscr{M}}$ captures task interaction relationships determined jointly by task-chain structure and target assignment.

Based on the definition of the Mission Layer, Algorithm 2 formalizes the construction of the Mission Layer network. During this process, tasks derived from operation loops whose length exceeds $L_{\max}$ are filtered out, since overly long loops generally imply higher execution complexity and lower stability. Accordingly, these tasks are pruned to ensure that the remaining Mission Layer nodes comply with the meta-task chain formation constraints and are more representative of practically executable missions.

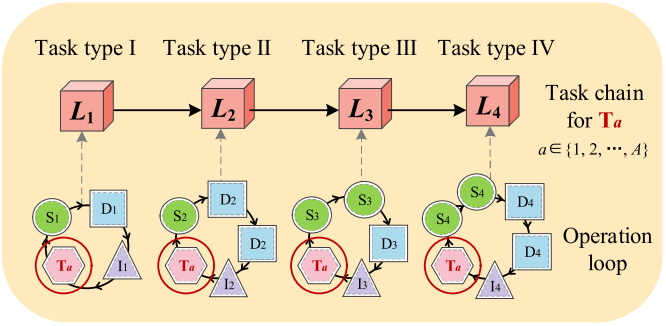
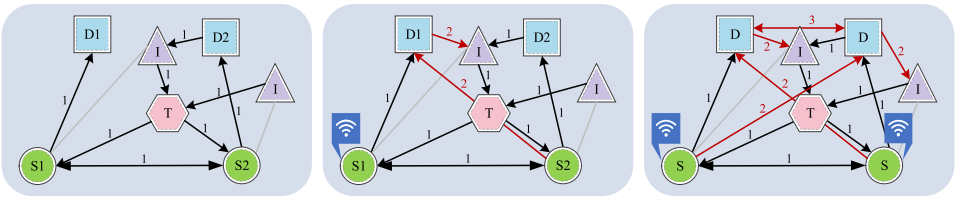


Fig. 5. Correspondence between meta-operation loop and mission linkage.



(a) $N_R = 0$ **(b)** $N_R = 1$ **(c)** $N_R = 2$

Fig. 4. Capability layer networks under different numbers of relay entities.

Algorithm 2
Mission layer network generation.

Input: Operation loop dictionary D_{loop}
Task chain rules R_{chain}
Maximum cycle length L_{max}
Output: Mission Layer Network $\mathcal{M}$
Initialize empty graph $\mathcal{M}$
foreach *operation_loop* (*idx, loop*) in D_{loop} **do**
  if *length_of_loop* $> L_{max}$ **then**
  continue
  end
  Identify the type of loop and its task type index t
  Compute *task_weight* based on *length_of_loop*
  Add node *idx* to $\mathcal{M}$
end
foreach (*src, tgt*) in $\mathcal{M}$, $src \neq tgt$ **do**
  if $R_{chain}(src_{type}, tgt_{type}) = True$ **then**
  Add directed *edge* ($src \rightarrow tgt$) in $\mathcal{M}$
end
return $\mathcal{M}$

2.2. TDMN-based USoS performance evaluation method

Disturbances acting upon USoS entities can induce topological changes in the Structure Layer, which subsequently affect the formation of operation loops and alter task-execution capabilities. In view of this cross-layer propagation mechanism, a TDMN-based evaluation method provides an effective and appropriate means for assessing USoS performance. The overall procedure of this evaluation method is illustrated in Fig. 6. Following this framework, USoS performance evaluation is conducted through the following three steps:

(1) Structure Layer Evaluation

In engineering, system availability is widely regarded as a fundamental indicator of system integrity. This perspective evaluates the overall health of an entity based on its operational availability. The functional integrity level of an entity $D(v_i)$ is typically expressed as Eq. (3):

$$D(v_i) = \frac{MTBCF}{MTBCF + T_{repair} + MLDT + MTTR} \tag{3}$$

where *MTBCF* denotes the mean time between critical failures; T_{repair} represents the fault detection time; *MLDT* denotes the fault isolation time; and *MTTR* is the mean time to repair, which includes both corrective and preventive maintenance durations.

(2) Capability Layer Evaluation

The following presents the capability evaluation method for USoS operation loops. Assuming that an operation loop $c_{y,l,a}$ contains the sensor node set V_y^S, the decider node set V_y^D, and the influencer node set V_y^I, the performance is evaluated using Eq. (4) [23]:

$$C(c_{y,l,a}) = \sum_{v_i \in V_y^S} D(v_i) \times \sum_{v_i \in V_y^D} D(v_i) \times \sum_{v_i \in V_y^I} D(v_i) \tag{4}$$

Where $c_{y,l,a}$ denotes the operation loop y of type l associated with target a. This formula quantifies the capability level of the operation loop by summing the functional capability levels involved in the loop.

Given that relay-based communication is inherently less stable than direct communication due to the cumulative failure probability across multiple hops and the stricter stability conditions observed in relay networks, USoS performance evaluation must consider not only the number of operation loops but also the difficulty of loop closure under multi-hop, relay-dependent conditions [39,40]. The overall capability of each operation loop can be further assessed by referencing the shortest length of its corresponding meta-operation loop by using Eq. (5):

$$F(c_{y,l,a}) = \frac{L_{min,l} \cdot C(c_{y,l,a})}{L(c_{y,l,a})} \tag{5}$$

ptypelassociatedwithtargeta.

where, $L(c_{y,l,a})$ denotes the total number of communication hops along all edges in operation loop $c_{y,l,a}$, and $L_{min,l}$ represents the minimal number of hops required to complete the meta-operation loop l.

All operation loops in the Capability Layer can be identified using a Breadth-First Search algorithm. In practice, each meta-operation loop corresponds to a specific system task. Accordingly, the Mission Layer network can be constructed based on the set of operation loops identified in the Capability Layer.

(3) Mission Layer Evaluation

Based on the capabilities of operation loops, the overall evaluation process in the Mission Layer is shown in Fig. 7. Since both task execution and task-chain formation in the USoS are organized around specific targets, the first step is to quantify the influence on each target. For target entities a, Average task-support level $\bar{E}_{l,a}$ for task type l to target a can be computed using Eq. (6):

$$\bar{E}_{l,a} = \frac{\sum_{y=1}^{n_{l,a}} F(c_{y,l,a})}{n_{l,a}} \tag{6}$$

where $n_{l,a}$ denotes the total number of tasks type l to target a. Building on the task-support level, it is also necessary to consider both the total number of meta-task chains associated with each target entity and their respective importance. Given TP_a denote the list of required meta-task chain of target a, the Influence Effect on target a can then be computed using Eq. (7):

$$R_a = \sum_{p=1}^{n_p} \omega_{p,a} \cdot (Ln_{a,p} \cdot \prod_{\mathscr{P}_p} \bar{E}_{l,a}) \cdot \delta(\mathscr{P}_p, TP_a) \tag{7}$$

where $\omega_{p,a}$ represents the importance weight of meta-task chain p for target a, and $Ln_{a,p}$ denotes the number of such chains associated with the target. The product of $Ln_{a,p}$ and $\prod_{\mathscr{P}_p} \bar{E}_{l,a}$ corresponds to the dashed components in Fig. 7, capturing the aggregated contribution of all meta-task chains linked to target a. The indicator function is used to determine whether the meta-task chain p is required for target a, as defined in Eq. (8):

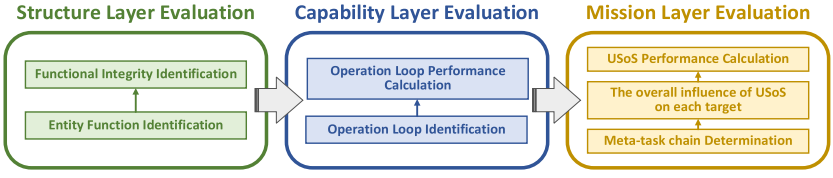


Fig. 6. The TDMN-based performance evaluation workflow.

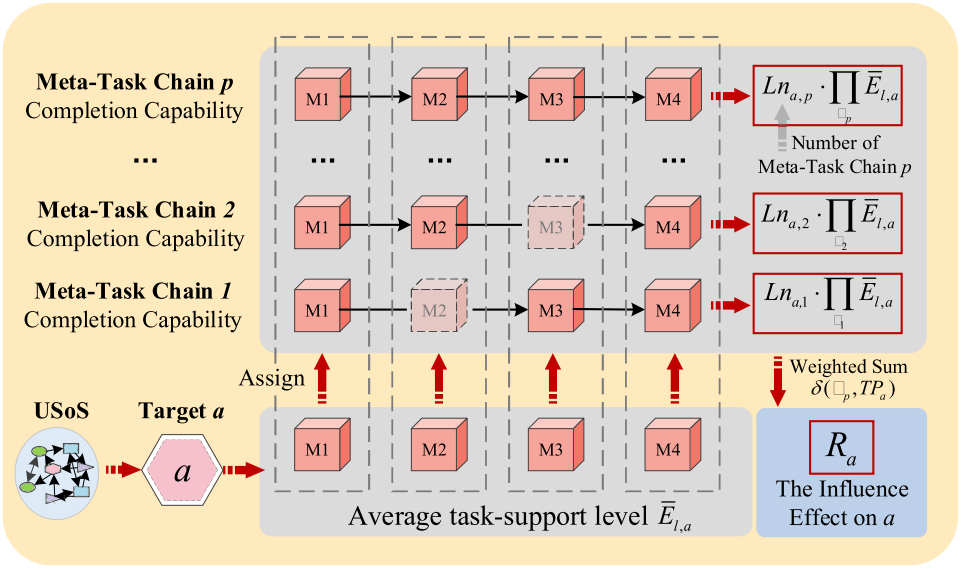


Fig. 7. Target influence effect evaluation process.

$$\delta\left(\mathscr{P}_p, TP_a\right) = \begin{cases} 1 & if\ \mathscr{P}_p \in TP_a \\ 0 & otherwise \end{cases} \tag{8}$$

Under real-world operational conditions, the influence effect exerted on target entities exhibits a clear diminishing marginal impact on overall USoS performance. Once the required baseline level of influence is achieved, further increases in influence effect contribute progressively less to performance improvement. Therefore, a logarithmic function is adopted as the mapping function for evaluating USoS performance. Moreover, because different targets have different operational priorities, the evaluation assigns higher weights to the influence effect exerted on high-priority targets. Based on these considerations, the overall USoS performance is computed using Eq. (9):

$$\psi_{SoS} = log_{\alpha_d}\left(1 + \sum_{a=1}^{A}\left(\lambda_a \cdot R_a\right)\right)\left(\alpha_d \geq 1\right) \tag{9}$$

where λ_a denotes the normalized importance weight of target a, and α_d represents the marginal decay factor of the influence effect; a larger value of α_d indicates a faster attenuation in the marginal contribution of the influence effect to USoS performance. It should be noted that the parameters $\omega_{p,a}$, λ_a and TP_a appearing in the above equations must be determined by the user based on expert judgment, prior operational experience, and the specific characteristics of the intended application scenario. There is no universally accepted standard for their assignment; thus, the determination of these parameters inherently involves a degree of subjectivity and context-dependence.

Through the TDMN, the structural and capability components of a USoS incorporating relay communications can be accurately characterized. Based on the proposed USoS performance evaluation method, we then design the recovery optimization method.

3. The bi-objective recovery optimization method

3.1. The problem of USoS recovery

3.1.1. Problem description

The recovery process of a USoS can be formulated as a combinatorial optimization problem that involves scheduling disabled entities and allocating limited recovery resources. Effective strategy formulation requires integrating the specific recovery needs of each entity with the feasible recovery measures. These measures may draw on internal resources—such as Redundant Backup, Structure Reconfiguration [31, 41], and Functional Switching—or external resources, including On-site Maintenance and Resource Allocation [42,43]. In practice, their applicability is constrained by resource availability and operational conditions. As a result, the selection of recovery measures must balance internal and external resource usage while accounting for time constraints and the elevated priority of relay entities. Recovery strategies therefore need to be adapted to scenario-specific requirements.

USoS recovery scenarios can be categorized into Immediate Recovery Scenarios and the Recovery Scenario after Damage. In Immediate Recovery Scenarios, the system operates under continuous interference with limited time available for intervention, making low–time-cost recovery measures more appropriate. In contrast, the Recovery Scenario after Damage offers relatively ample recovery time because immediate countermeasures are not required. Accordingly, this study focuses on four recovery measures applicable to the Recovery Scenario after Damage: Functional Switching, Resource Allocation, Redundant Backup, and On-site Maintenance. A schematic representation of these measures is provided in Fig. 8.

- **Functional Switching.** When an entity's current functional payload fails due to interference, it switches to other intact functional payloads.
- **Resource Allocation.** Deployment of replacement entities from the nearest support site equipped with the required resources to substitute damaged entities.
- **Redundant Backup.** Preconfigured spare parts for vulnerable core entities, immediately activated after disruption to maintain mission capability.
- **On-site Maintenance.** Dispatch of support vehicles from support sites to retrieve damaged entities for repair, after which they return to the mission area.

For the Recovery Scenario after Damage, and considering common constraints in optimization problems [17,18,44], the following assumptions are made:

1) Two damage levels are defined. Complete damage, where all capabilities of the entity instantly drop to zero and Functional Switching is unavailable; partial damage, where the current function capability of the entity drops to zero but other functional payloads remain intact, allowing Functional Switching.

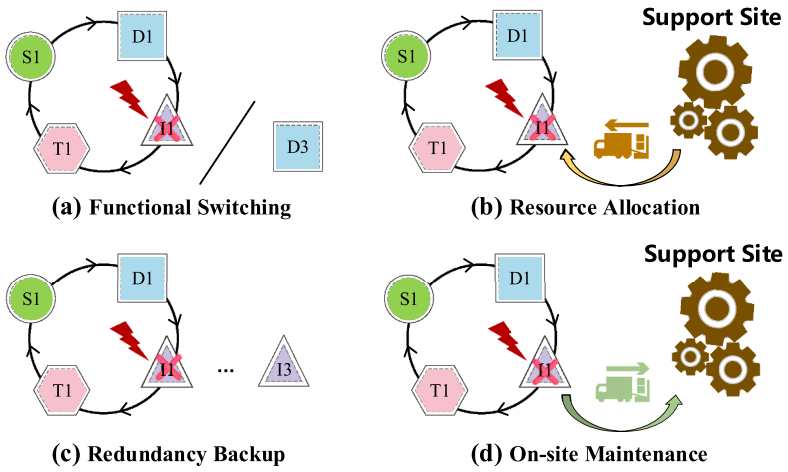


Fig. 8. Four types of recovery operations in USoS.

2) Only entity damage is considered, without independent link disturbances. A link in the Structure Layer is regarded as damaged if either endpoint entity is damaged; the link is automatically restored once the entities recover. Disturbances affecting inter-entity communication or the information forwarding functions of relay entities are not considered.
3) All entity information is assumed to be known, including payload configuration, the presence of relay modules, current capability states, and geographic locations.
4) If an operational loop is closed, it is considered functionally achievable, meaning that functional interaction and task completion can be realized. The cumulative latency and potential signal degradation introduced by multi-hop relay transmission are not additionally considered in evaluating USoS performance.

Furthermore, considering the impacts of usage environment, human factors, and entity importance on the execution quality of recovery measures, a recovery measures quality ladder concept—similar to tiered electricity pricing [33]—is introduced during recovery strategy optimization to balance recovery effectiveness and efficiency. This mechanism also provides a means to allocate additional recovery resources to critical nodes, such as relay entities and their neighboring nodes.

3.1.2. *Optimization objective analysis*

The optimization of recovery strategies requires balancing the macroscopic resilience characteristics of the USoS with the operational efficiency of microscopic recovery measures. To this end, two evaluation metrics are employed: cumulative performance loss resilience and total recovery time cost. The former, derived from resilience-curve analysis and TDMN-based performance evaluation, quantifies capability restoration while capturing the effects of relay communication on resilience. The latter measures the operational efficiency of recovery measures by aggregating their execution durations. Together, these metrics form a multidimensional basis for evaluating the effectiveness and efficiency of recovery strategies.

(1) Cumulative performance loss resilience

Resilience is often assessed through performance curves that characterize system behavior during disturbance and recovery. Representative quantification methods include the resilience triangle model [2] and slope-based metrics [31,45], while additional studies incorporate communication constraints [46], information-exchange capabilities [11], and recovery mechanisms such as resource backup and structural reconfiguration [26] to demonstrate the influence of topology and support resources on USoS resilience [47,48]. Among these approaches, the resilience triangle provides a concise representation of cumulative performance loss and aligns with the discrete, stagewise degradation and restoration typical of USoS; it is therefore adopted as the basis for resilience quantification. As shown in Fig. 9, the interval from disturbance initiation to recovery initiation defines the Immediate Recovery Scenario. The subsequent period until the recovery completion time constitutes the Recovery Scenario after Damage. During this stage, the performance curve increases stepwise as recovery measures are applied, reflecting the discrete nature of the process. The black curve denotes system performance over time, and the shaded region above it represents performance loss. A smaller shaded area indicates sustained higher performance, demonstrating the effectiveness of the recovery strategy. The next section formalizes this shaded area to derive cumulative performance loss resilience.

Cumulative performance loss resilience measures how long the USoS operates under degraded performance, thus reflecting the effectiveness of the recovery strategy. We use the TDMN-based evaluation method as the performance metric for resilience calculation. The first step is normalizing the performance of USoS as shown in Eq. (10):

$$P(T_r) = \frac{\psi_{SoS}(T_r)}{\psi_{SoS}(T_0)} \tag{10}$$

where T_r denotes the current time and T_0 denotes the time before any damage occurs to the USoS. And then, resilience is evaluated using the resilience triangle model, whose general formulation is given by Eq. (11):

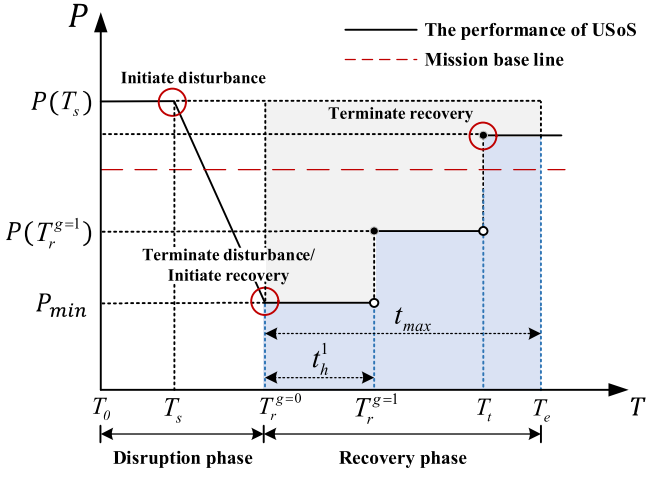


Fig. 9. Resilience curve of USoS.

$$\mathscr{R} = \int_{T_s}^{T_e} (P(T_s) - P(T))\mathrm{d}T \tag{11}$$

where $\mathscr{R}$ represents the cumulative performance loss resilience, with smaller values indicating higher resilience. $P(T_s)$ represents the initial performance of USoS. T_e represents the recovery termination time.

In practical applications, this metric requires normalization. Considering that the Recovery Scenario after Damage involves combinatorial optimization and discrete performance restoration, the integral can be approximated by a summation, resulting in the cumulative performance loss resilience formula shown in Eq. (12):

$$\mathscr{R}_{SoS} = \frac{\sum_{g=0}^{G} \left(P(T_s) - P\left(T_r^g\right)\right) \times t_h^{g+1}}{t_{max} \times P(T_s)} \tag{12}$$

where $T_r^0 < T_r^1 < \ldots < T_r^g < \ldots < T_e$, $T_e = T_r^{g=0} + t_{max}$, and $t_h^{g+1} = T_r^{g+1} - T_r^g$. t_{max} represents the maximum recovery cost allowed for USoS. If the system completes recovery before T_e, then $T_t \leq T_e$; if the recovery process is slow, $T_t > T_e$.

(2) Total recovery time cost

Although the Recovery Scenario after Damage is less sensitive to time, it remains necessary to calculate the total recovery time cost to evaluate the effectiveness of different recovery strategies. The time cost of each recovery measure comprises operation time cost F_{cto} and mobilization time cost F_{mto}. The total recovery time is computed as shown in Eq. (13):

$$TF_{SoS} = F_{cto} + F_{mto} \tag{13}$$

The specific methods for calculating operation and mobilization times will be detailed in the Section 3.2.

Based on the previously proposed the Recovery Scenario after Damage and objective functions, Fig. 10 illustrates an example recovery process of a USoS comprising eight entities to demonstrate the impact of different metrics on resilience evaluation. In this example, the initial capability of each entity is set to 1. Entities 2, 3, and 6 are damaged and restored via Redundant Backup, Resource Allocation, and On-site Maintenance, respectively.

3.2. Mathematical model

The mathematical model for USoS recovery after damage is developed based on the following scenario. After experiencing disturbances, several entities are damaged, affecting the successful completion of missions. During recovery, a sequence of support sites providing maintenance and resource allocation is denoted by $K_{SOS} = \{K_k | k = 1, 2, \ldots, n_k\}$, and the distance from entity v_i to each support site is $DL_i = \{d_{i,k} | k = 1, 2, \ldots, n_k\}$. Each recovery measure performed by USoS can select from multiple recover quality levels, each associated with a recovery-effect coefficient and a recovery-time coefficient $Q_{SOS} = \{(Qe_q, Qt_q) | q = 1, 2, \ldots, n_q\}$, describing the degree of capability restoration and the time required for that level. The minimum number of hops required to establish communication between entities is represented by $HOP_{SOS} = \left\{hop_{i_1,i_2} | i_1, i_2 = 1, 2, \ldots, n and i_1 \neq i_2\right\}$. The four types of recovery measures are indexed by j, with values from 0 to 4 representing No Recovery, Redundant Backup, Functional Switching, Resource Allocation, and On-site Maintenance, respectively. Since the proposed optimization method is invariant to parameter units, all subsequent parameters in the mathematical model are treated as dimensionless quantities to enhance generality and scalability.

(1) Optimization Objectives

The objective functions of the optimization problem correspond to the previously defined cumulative performance loss resilience $\mathscr{R}_{SoS}$ and the total recovery time cost TF_{SoS}, which can be expressed as Eq. (14):

$$\min\{\mathscr{R}_{SoS}, TF_{SoS}\} \tag{14}$$

(2) Decision Variables

Let the decision variables in the USoS recovery strategy optimization model be denoted as $x_{i,j,k}$ and $z_{i,q}$, where:

$$x_{i,j,k} = \begin{cases} 1 & if entity i selects recovery measure j and support site k \\ 0 & otherwise \end{cases}, \tag{15}$$

$$z_{i,q} = \begin{cases} 1 & if entity i selects recovery quality level q \\ 0 & otherwise \end{cases}. \tag{16}$$

The decision variables described in Eq. (15) and Eq. (16) collectively represent a complete recovery strategy within the USoS recovery optimization model. The detailed encoding and decoding procedures for these variables are presented in Section 3.3.2.

(3) Constraints

We incorporate various constraints—including logical, resource, temporal, and task-related constraints—to abstract the USoS recovery process in a structured manner, thereby laying the foundation for subsequent algorithm design and solution. Constraint (17) enforces the mutual exclusivity of different types of recovery measures, ensuring that each damaged entity can undergo at most one recovery measure during

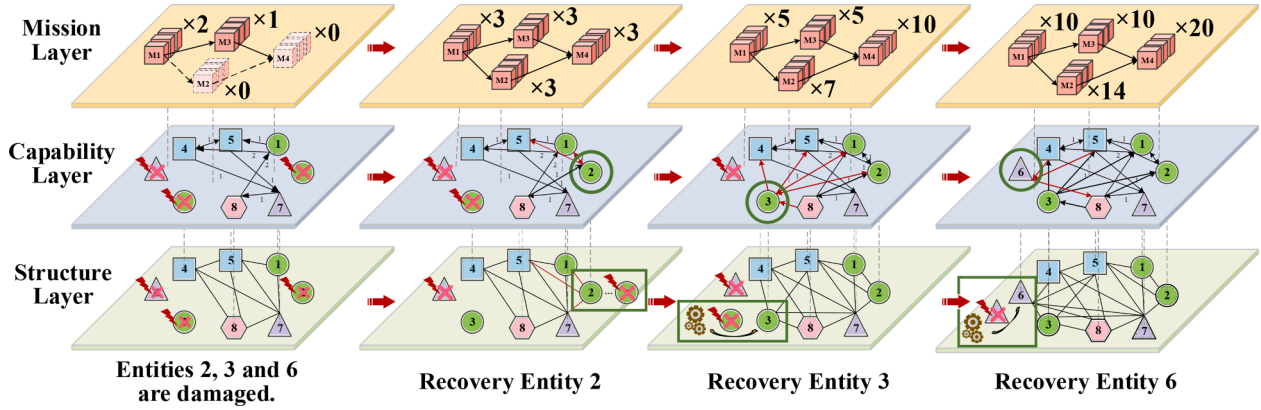


Fig. 10. USoS recovery processes.

a single recovery operation. Constraint (18) restricts each entity's recovery measure to involve no more than one support site, and ensures that the total amount of resources allocated from each support site does not exceed its storage capacity TR_k. Constraint (19) enforces that the number of backup units used for recovery cannot exceed the configured backup quantity RB. Constraint (20) requires that only one level of recovery quality be selected for each recovery measure to avoid conflicts with the problem setting.

$$\sum_{k=1}^{n_k}\sum_{j=0}^{4} x_{i,j,k} \le 1, \forall i \tag{17}$$

$$\sum_{i=1}^{n}\sum_{j=0}^{4} x_{i,j,k} \le TR_k, \forall k \tag{18}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{n_k} x_{i,1,k} \le RB \tag{19}$$

$$\sum_{q=1}^{Q} z_{i,q} \le 1, \forall i \tag{20}$$

During the USoS recovery process, the restored performance of an entity is determined by the selected recovery measure and its associated recovery quality level. Since only functional switching replaces the currently damaged function with other onboard payloads already available on the platform, and the newly activated capability level may also be affected by previous strikes, let τ denote the degradation coefficient caused by Functional Switching [49]. Then the functional integrity level after recovery of entity i. D_i^* can be calculated using Eq. (21).

$$D_i^* = \begin{cases} D_i, \sum_{j=0}^{n}\sum_{k=1}^{s} x_{i,j,k} = 0 \\ D_i + (1-D_i)\cdot\sum_{q=1}^{n_q} z_{i,q} Qe_q \cdot \tau, \sum_{j=0}^{4}\sum_{k=1}^{n_k} x_{i,j,k} = 1 \wedge \sum_{k=1}^{n_k} x_{i,2,k} = 1 \\ D_i + (1-D_i)\cdot\sum_{q=1}^{n_q} z_{i,q} Qe_q, \sum_{j=0}^{4}\sum_{k=1}^{n_k} x_{i,j,k} = 1 \wedge \sum_{k=1}^{n_k} x_{i,2,k} \ne 1 \end{cases} \tag{21}$$

Under the scenario assumptions, the spatial locations of entities and support sites, the execution time of each recovery measure, and the mobilization time of each entity are known. Support vehicles move at speed v_s, while all entities move at a speed of v_m. Thus, the operation time cost F_{otc} and mobilization time cost F_{mtc} of a recovery plan are computed using Eq. (22) and Eq. (23).

$$F_{otc} = \sum_{i=1}^{n}\sum_{k=1}^{n_k}\sum_{j=0}^{4}\left(x_{i,j,k}\cdot F_{spt,j}\cdot\sum_{q=1}^{n_q} z_{i,q} Qt_q\right) \tag{22}$$

$$F_{mtc} = \sum_{i=1}^{n}\sum_{k=1}^{n_k}\left\{\sum_{j=3}^{4}\left(\frac{x_{i,j,k}\cdot d_{i,k}}{v_m}\right) + \frac{x_{i,4,k}\cdot d_{i,k}}{v_s}\right\} \tag{23}$$

In terms of task-related constraints, to ensure that the USoS possesses sufficient capability to complete its missions, Constraint (24) requires that all meta-task chain must be feasible for every target. Constraint (25) ensures that the post-recovery system performance exceeds the predefined mission baseline, reflecting the effectiveness of the recovery strategy. Constraint (26) imposes an upper limit on the total recovery time, corresponding to the maximum allowable time cost for completing recovery after damage.

$$\prod_{l\in\mathscr{P}_p}\overline{E}_{l,a} > 0 \tag{24}$$

$$P(T_t) > P_{BL} \tag{25}$$

$$t_R \le t_{max} \tag{26}$$

In addition, according to the TDMN-based performance evaluation method, any newly generated operation loop resulting from node recovery or functional transformation must satisfy Constraint (27) and Constraint (28). These constraints respectively require that the relay-based communication hop count of a new loop shall not exceed the prescribed upper bound $HOP_{\max}$, and that the total loop length must remain below the predefined total loop length $TH_{\max}$. This hop limitation is imposed because multi-hop relaying causes communication delay, reliability, and throughput to degrade sharply with increasing hops, making overly long relay paths unable to satisfy the timeliness required by OODA-based operations [50].

$$hop_{i_1,i_2} \le HOP_{\max}, \forall i_1, i_2 = 1, 2, \ldots, nandi_1 \ne i_2 \tag{27}$$

$$TH(l_{new}) \le TH_{\max} \tag{28}$$

With the mathematical model for USoS recovery strategy optimization fully established, the value domains of all parameters and the calculation procedures for optimization objectives are now clearly defined. Based on this model, a corresponding solution algorithm will be developed in the next section.

3.3. Description of BMRMFO

3.3.1. Algorithm procedure

Building on the mathematical formulation of the Recovery Scenario after Damage, we develop BMRMFO, an optimization algorithm tailored for USoS recovery under relay communication. Compared with standard recovery settings, relay-enabled USoS exhibit power-law structural features, deeper hierarchical coupling, and stronger nonlinear relationships between network scale and system performance—characteristics that require extending the classical MFO framework [51,52].

MFO, inspired by moth navigation, performs Pareto-based ranking of individuals as moths and flames, and updates solutions through spiral movements toward high-quality flames. Although MFO offers strong global exploration and boundary coverage for discontinuous Pareto fronts, directly applying it to USoS recovery still faces issues such as a sparse feasible space and premature convergence [53–55]. Therefore, BMRMFO first introduces a discrete encoding consistent with the proposed recovery optimization model, enabling the search to proceed within the true feasible set. On this basis, two auxiliary enhancements are adopted to further improve the algorithm–model compatibility: (1) a maintenance-rule-based initialization strategy that enhances the quality and feasibility of the initial population, and (2) a local reinforcement mechanism that applies controlled mutations to selected individuals, strengthening local search and reducing the risk of convergence to suboptimal regions. The algorithm flow is illustrated in the Fig. 11.

3.3.2. Initialization

The decision variables of the USoS recovery strategy consist of: a three-dimensional matrix $x_{i,j,k}$ representing the sequence of recovery measures, and a two-dimensional matrix $z_{i,q}$ representing the quality levels of recovery operations. The specific meanings of each coordinate in the decision variables have been detailed previously.

The encoding and decoding procedure of BMRMFO is illustrated in Fig. 12. Under the Recovery Scenario after Damage, the indices, types, and relevant parameters of damaged entities are first extracted. Integrating the information on support sites and redundant resources, each recovery action is specified by determining its target entity, recovery measure, and execution site. These actions are encoded by assigning value 1 to the corresponding coordinates in a three-dimensional matrix. Subsequently, based on the quality-level rules, a single coordinate in a two-dimensional matrix is set to 1 to indicate the selected recovery

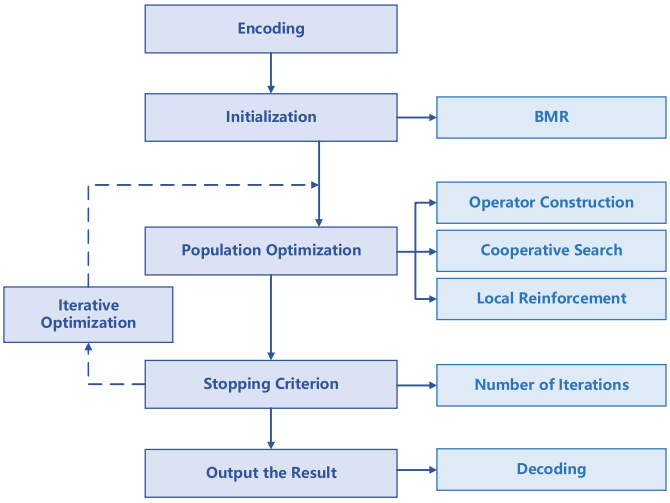


Fig. 11. The flowchart of BMRMFO.

quality for each action. This encoding scheme transforms the post-damage state of the USoS into a solution representation suitable for optimization.

To generate feasible and diverse initial solutions, a maintenance-rule-based initialization strategy BMR is developed. Formulating a recovery strategy involves three coupled decisions—selecting recovery measures, assigning support sites, and determining recovery quality levels—requiring an initialization method that balances practical recovery logic with population diversity. After a disturbance, the damage state and attributes of each affected entity are assessed, and a recovery decision sequence is constructed to prioritize the allocation of support resources. Guided by this sequence, the algorithm first checks the availability of redundant backups and uses them preferentially; if the entity is only partially damaged, a Functional Switch is attempted. Otherwise, resources are dispatched to the nearest support site capable of replenishment; when no site has sufficient resources, the nearest site is assigned to perform On-site Maintenance. This procedure generates an initial population that follows the maintenance-rule-based action selection while maintaining diversity through randomized recovery measures quality assignment, and the complete workflow is summarized in Algorithm 3.

3.3.3. *Algorithm procedure*

After population initialization, the algorithm proceeds to an iterative optimization phase consisting of three components: moth–flame position updating, local reinforcement, and Pareto archive maintenance. These components collectively steer the population from the initial feasible region toward the Pareto-optimal front. The complete algorithmic workflow is depicted in Fig. 13.

At the beginning of each iteration, the algorithm adaptively adjusts the number of flames according to the iteration index. A larger flame set is preserved in the early phase to enhance population diversity and global exploration, whereas a linearly decreasing schedule is applied as iterations progress to strengthen convergence. The flame reduction process follows Eq. (29):

$$N_f = round\left(\frac{(N - h_{iter})(N-1)}{H}\right) \tag{29}$$

where N_f denotes the flame count at iteration, N is the initial population size, h_{iter} is the current iteration count, and H is the total number of iterations. A subset of non-dominated solutions from the current Pareto front is then selected to form the flame population. The remaining moths identify superior flames via binary tournament selection and generate offspring through crossover–replacement operations along key decision dimensions, as illustrated by the red component of the population optimization in Fig. 13. This updating process may yield infeasible individuals—most commonly those violating support-resource limits. Therefore, every new solution must undergo a feasibility check to ensure compliance with structural and resource constraints.

Step 2. Local Reinforcement

Algorithm 3
BMR Initial population generation.

Input: Damaged entities = $\{e_1, e_2, \ldots, e_n\}$

Redundant backups RB
Backups in support site k, TR_k
Dimensions of S_1: $n \times 5 \times p$
Dimensions of S_2: $n \times q$
Output: Solution set $S = \{S_1, S_2\}$ where S_1 and S_2 are updated matrices
Initialize $S_1 \leftarrow 0$ matrix of size $n \times m \times p$
Initialize $S_2 \leftarrow 0$ matrix of size $n \times q$
Generate $V \leftarrow$ recovery decision sequence
foreach $u_i \in V$ **do**
  $q_i \leftarrow 0$
  if Redundant backups utilized in the recovery strategy $< RB$ **then**
    Set $S_1(i, 1, 0) = 1$
  end
  else if u_i *is partially damaged* **then**
    Set $S_1(i, 2, 0) = 1$
  end
  else if $TR_{k,type(ui)} > 1$ **then**
    Set $S_1(i, 3, k) = 1$
  end
  else
    Set $S_1(i, 4, 0) = 1$
  end
  Randomly select z such that $S_2(i, z) = 1$
  Set $S_2(i, z) = 1$
  $S \leftarrow S \cup \{S_1, S_2\}$
end
return S

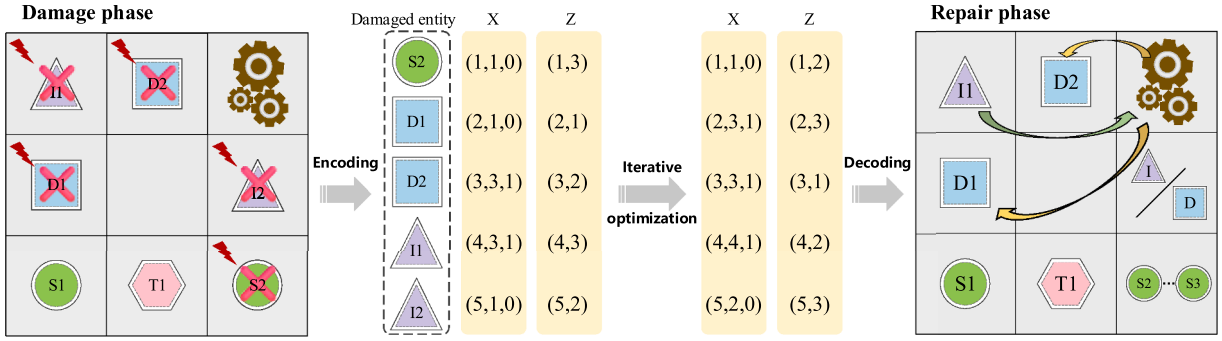


Fig. 12. Encoding and decoding process of BMRMFO.

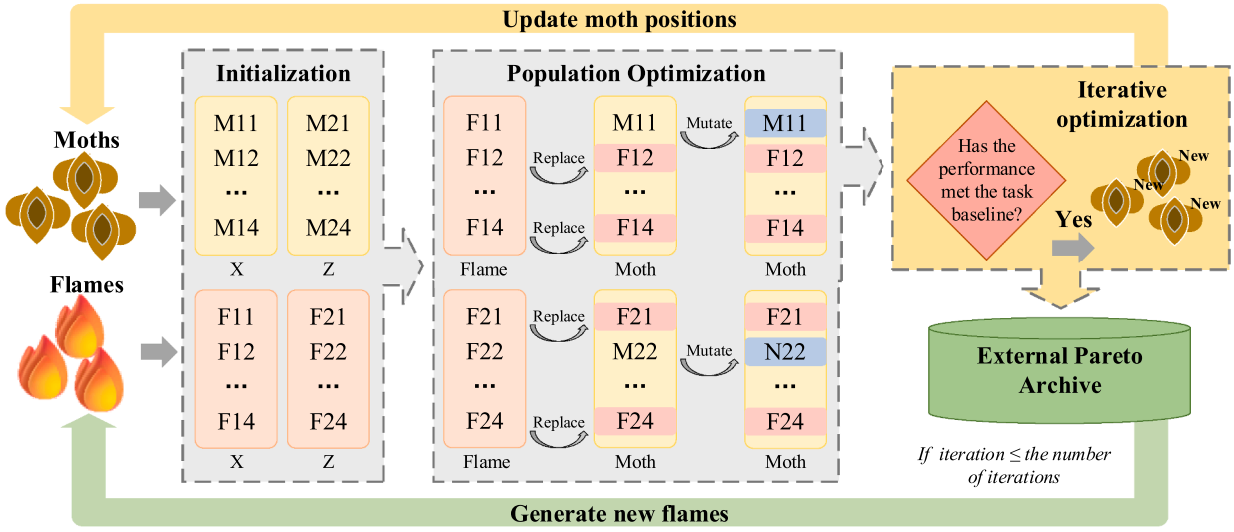


Fig. 13. Optimization process of BMRMFO.

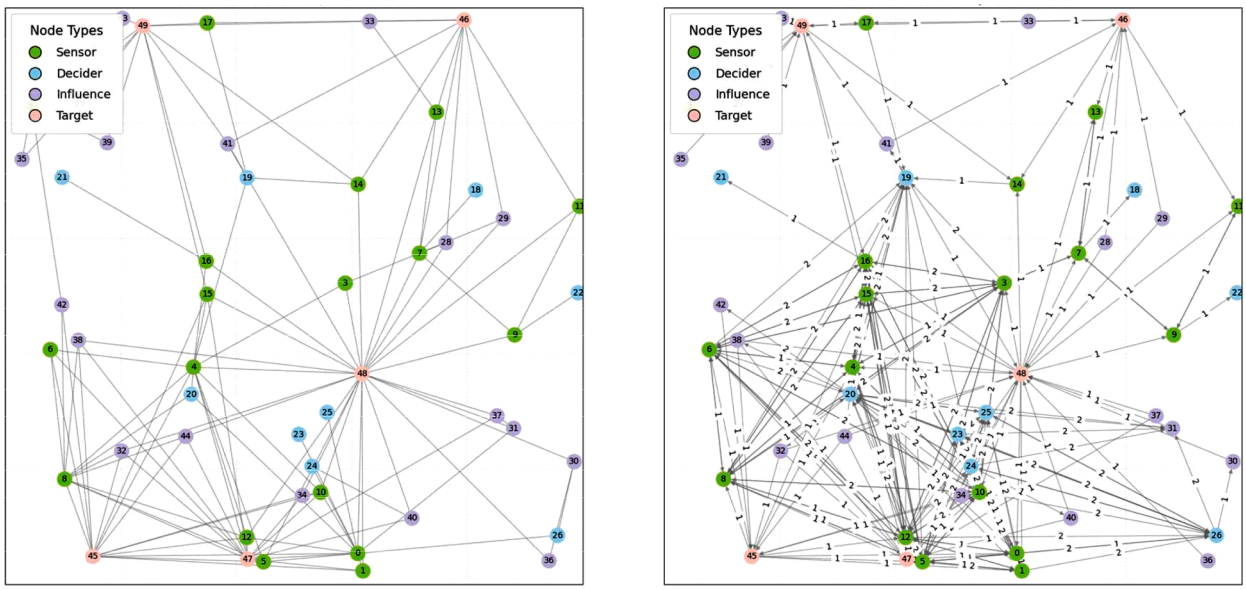


Fig. 14. Structure layer and capability layer networks of USoS under configuration cases 5.

To enhance solution quality while preserving population diversity, a selected subset of moths undergoes local reinforcement, corresponding to the blue shaded region in Fig. 13 generated by partial mutation during population optimization. For each chosen individual, the values of several decision variables are randomly perturbed to generate candidate solutions. These candidates are then evaluated against the original individual using Pareto-dominance criteria across both objective functions. If a superior candidate is identified within the prescribed attempt limit, it replaces the original; otherwise, the original solution is retained. This reinforcement mechanism enables focused exploitation of the local search space while ensuring constraint feasibility, thereby mitigating premature convergence and improving overall optimization performance. The detailed procedure is presented in Algorithm 4.

Step 3. Pareto Archive Update

After the local reinforcement phase, the current population is merged with the flame population from the previous iteration to form a unified candidate set, as illustrated by the yellow and green components in Fig. 13. This set undergoes non-dominated sorting, where individuals are ranked using dominance and crowding-distance criteria, followed by an update of the external Pareto archive. The resulting non-dominated individuals constitute the flame population for the next iteration and serve as reference points for subsequent moth updates. This iterative optimization cycle repeats until the maximum iteration count is reached, at which point the algorithm outputs the Pareto front representing the trade-off solutions between resilience performance and recovery time cost.

Following the above algorithmic steps, an optimal recovery strategy can be derived that simultaneously balances cumulative performance loss resilience and total recovery time cost, based on the current USoS composition, available support resources, support site locations, and damaged conditions.

Algorithm 4
Local reinforcement procedure.

Input: Population Mo_t
Mutation rate g
Acceptance factor $\mathscr{E}$
Output: Updated population Mo_{new}
$Mo_{new} = Mo_t$
for *iter* ← 1 **to** *round* ($popnum \cdot g$) **do**
  Estimate objective bounds ($\max f_{iter}$, $\min f_{iter}$) over all *mo*
  for *attempt* ← 1 to *max_attempts* **do**
    Randomly select index *mo* from Mo_t
    Propose modified solution *mo_new* by applying mutation to *mo*
    Evaluate new objectives (f_1, f_2)
    if *mo_new* satisfies $\mathscr{E}$-*dominance* over *mo* then
      Replace *mo* with *mo_new* in Mo_{new}
      Replace *mo* from Mo_t; **break**
    end
  end
end
return Mo_{new}

4. Case study

4.1. Case parameter settings

To verify the feasibility and effectiveness of the proposed TDMN and BMRMFO in improving USoS resilience, a series of comparative experiments is conducted using a representative case study. The evaluation includes two complementary components: (i) validating TDMN's ability to capture relay-induced structural impacts and a loop-based metric is constructed and compared with the TDMN-based metric, demonstrating the necessity of incorporating task chains into performance evaluation; (ii) benchmarking BMRMFO against classical multi-objective algorithms and conventional recovery-strategy methods, supplemented by significance testing and relay-priority analysis. The case parameters are configured by referencing representative prior studies [2,18] and commonly adopted USoS scales.

Among entities of USoS, three sensors are equipped with relay communication model, and a redundancy set is included comprising two sensors, one decider, and one influencer. The communication range of each entity is set to $r_c = 40$, while the observation range of sensors and the strike range of influencers are set to $r_s = 50$ and $r_l = 50$, respectively. To evaluate the performance of BMRMFO, we varied three parameters: the total number of USoS entities, the number of support sites, and the intensity of disruption, thereby constructing a range of experimental instances. Detailed configuration information of the USoS is provided in Table 1.

The support sites in the USoS are capable of executing Resource Allocation or On-site Maintenance according to system requests. The rescue vehicles operate at a speed of $v_s = 25$, while all entities move at a speed of $v_m = 20$.

Different backup resources are allocated to each support site, with relevant information provided in Table 2. In addition, various disturbance intensities are configured in the case study to match the available support resources. Specifically, for total number of support sites is 2, 4 and 6, the selected support sites ID are 5 and 6, 1–4, and 1–6 respectively.

Table 1
USoS Configuration cases.

ID	Total Number	Sensor	Decider	Influencer	Target
1	30	10	5	10	5
2	35	12	6	12	5
3	40	14	7	14	5
4	45	16	8	16	5
5	50	18	9	18	5
6	55	20	10	20	5
7	60	22	11	22	5
8	65	24	12	24	5
9	70	26	13	26	5

Table 2
Support site configuration parameters.

ID	Position	Sensor	Decider	Influencer
1	(25, 75)	2	1	1
2	(25, 25)	1	1	2
3	(75, 75)	2	2	0
4	(75, 25)	0	1	2
5	(50, 50)	2	1	1
6	(30, 70)	1	2	2

The compositions of the three levels of total damaged ratio d_r are listed in Table 3. The coupling between different disturbance levels and support resource configurations enables the construction of diverse scenario, allowing a more comprehensive evaluation of the proposed method's generalizability and effectiveness.

This case study set basic operation times of $F_{spt,1} = 8$, $F_{spt,2} = 10$, $F_{spt,3} = 12$ and $F_{spt,4} = 15$ respectively. Additionally, the recovery measures correspond to different cost increase factors and function degradation coefficients. The cost increase factors are $Qe_1 = 3$, $Qe_2 = 2$ and $Qe_3 = 1$ while the function degradation coefficients are $Qt_1 = 1$, $Qt_2 = 0.8$ and $Qt_3 = 0.5$. At the Mission Layer, each meta-operation loop corresponds to a task. USoS is required to execute two meta-task chains for each target: $\{L_1 \rightarrow L_2 \rightarrow L_4\}$ and $\{L_1 \rightarrow L_3 \rightarrow L_4\}$. Finally, the communication interactions in the Structure layer are determined based on the modeling method proposed in [37]. The model generation parameters and other case-specific parameters are listed in Table 4. Since BMRMFO is independent of physical units, all case parameters (e.g., length, speeds, resource capacities) are normalized into dimensionless form. The section begins with an overview of the case background.

4.2. TDMN Validation analysis

4.2.1. Relay impact validation and analysis

To verify whether the proposed TDMN can effectively characterize the performance-gain features of relay communication systems, this section conducts comparative experiments by progressively introducing relay entities and analyzing the structural evolution of the Capability Layer within TDMN. Using the configuration defined in configuration cases 5, the number of relay entities N_R is increased from 0 to 30. For each value of N_R, fifteen Monte Carlo trials are conducted by repeatedly reconstructing the Structure Layer to mitigate stochastic fluctuations. To further distinguish the roles of different information-flow mechanisms in the Capability Layer, four communication configurations are examined, each corresponding to a distinct subset of admissible information flows defined in Section 2.2, including allowing all information flows, disabling $\{S \leftrightarrow S\}$, disabling $\{D \leftrightarrow D\}$, and disabling both $\{S \leftrightarrow S\}$ and $\{D \leftrightarrow D\}$. Through these configurations, we assess whether TDMN can consistently capture the structural characteristics induced by relay deployment under different capability-formation constraints.

For each trial, we recorded multiple structural indicators including: the number of feasible operation loops; natural connectivity, measuring global robustness and the average shortest-path hop count under relay-enabled routing, and performed global normalization on all collected data. Fig. 15 report the mean values of each structural metric, with shaded regions showing the observed variability.

From Fig. 15, it is apparent that the number of operation loops

Table 3
Disruption intensity parameters.

d_r	d_c	d_p
30	20	10
50	30	20
70	40	30

Table 4
Network generation parameters.

Parameter	Value	Definition
r	10^2	Side length of square region
Sq	10^4	Region area
β	0.9	An attachment parameter represents weights of degree and distance
g	1	Number of communication links a new entity makes with existing entities when it joins the swarm
s'	0.5	A constant to adjust the midpoint of distance function $Fun(d_{i2,i1})$
HOP_{max}	2	Maximum communication hop limit
TH_{max}	8	Maximum operation loop length limit

Based on the above case background, taking USoS configuration cases 5 as an example, the resulting Structure Layer network is shown in Fig. 14 (a), and the corresponding Capability Layer network is shown in Fig. 14 (b).

increases rapidly with the growth of relay entities N_R. Compared with the scale expansion reflected by the number of operation loops, natural connectivity and average shortest-path length capture the evolution of connectivity quality in the capability network. The curves for "All information flows are permitted" and "Disable {D↔D}" nearly overlap, diverging only slightly in the high-N_R. region. This indicates that the primary sensing–decision–influence chain dominates the spectral connectivity of the network, while additional {D↔D} links—though substantially increasing the number of loops—contribute only marginally to global reachability and spectral structure. The average shortest-path length shows an initial increase followed by a pronounced decline and stabilization. With few relays, expansion of the largest connected component temporarily lengthens paths, whereas further relay deployment introduces cross-region shortcuts that shorten paths and drive convergence to a low stable level, consistent with small-world behavior. These results indicate that TDMN can consistently and realistically capture the impact of relay deployment on the connectivity quality of the capability network under different communication configurations.

To verify that this trend reflects TDMN's ability to capture relay-driven structural evolution rather than incidental topological accumulation, a power-law fitting analysis is conducted, and the corresponding results are summarized in Table 5. The results show that the "All information flows are permitted" exhibits the largest power-law exponent, with α=1.190 for configuration cases 5 and α=1.143 for configuration cases 9, followed by "Disable {D↔D}" with α=0.965 and 0.93 and "Disable {S↔S}" with α=0.966 and 0.901; the most restrictive setting, "Disable {S↔S} and {D↔D}," yields the smallest exponents. These numerical differences indicate that when communication constraints are minimal, the Capability Layer experiences the strongest combinatorial expansion, whereas restricting certain information flows moderates the growth rate but still maintains clear nonlinear scalability. And all configurations achieve very high goodness-of-fit values ($R^2 \geq 0.973$), which result further demonstrate that TDMN responds in a consistent and discriminative manner to different information-flow rules, rather than producing uniform or degenerate structural growth patterns.

Overall, the experimental results confirm that TDMN can stably, consistently, and discriminatively characterize relay-induced structural evolution, validating its effectiveness in modeling performance gains in relay-enabled USoS.

4.2.2. *Comparative validation of the TDMN-based metric*

Based on the preceding discussion on task chains, it can be observed that system task completion capability is not formed through the linear accumulation of individual capabilities or localized structural improvements, but instead depends on the coordinated evolution of multiple capability elements at the task chain-level. Consequently, whether a task-oriented evaluation method can faithfully reflect the formation and activation of task chains becomes a critical criterion for assessing its effectiveness. To further examine whether the proposed TDMN model can faithfully capture task-chain requirements, we conducted a

Table 5
Power-law fitting results of operation-loop growth.

Cases ID	Communication configuration	α	R^2
5	Disable {S↔S} and {D↔D}	0.749	0.995
	Disable {D↔D}	0.965	0.998
	Disable {S↔S}	0.966	0.990
	All information-flows are permitted	1.190	0.995
9	Disable {S↔S} and {D↔D}	0.718	0.989
	Disable {D↔D}	0.939	0.993
	Disable {S↔S}	0.901	0.973
	All information-flows are permitted	1.143	0.988

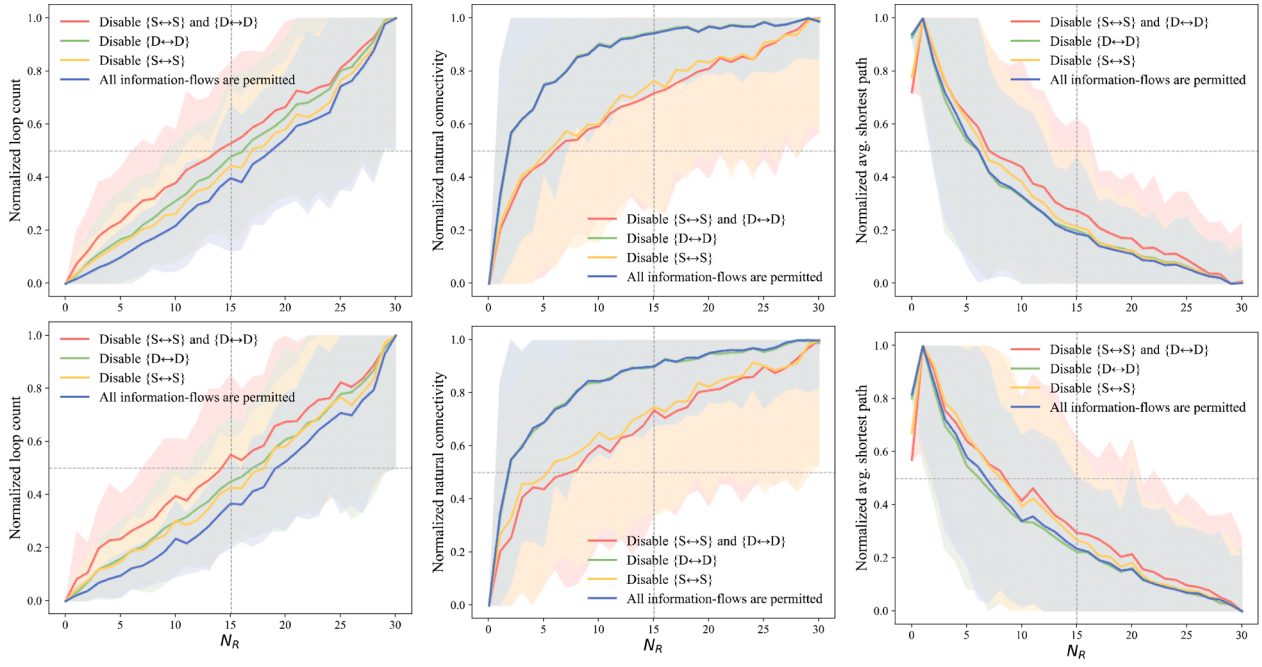


Fig. 15. Number of operation loops under different relay communication system.

comparative adaptability analysis under three representative USoS scenarios: configuration cases 2, configuration cases 5, and 9. Under identical damage conditions and identical recovery sequences, the evolution of system performance is evaluated using both the TDMN-based metric and a traditional loop-based metric.

For the loop-based metric, the contribution of meta-operation loops is first computed using Eq. (5). The influence effect on target a R_a^{loop} is then obtained following Eq. (30):

$$R_a^{loop} = \sum_{l=1}^{n_l} \frac{\omega_{l,a} \cdot \sum_{y}^{n_{l,a}} F(c_{y,l,a})}{n_{l,a}} \tag{30}$$

where $\omega_{l,a}$ denotes the importance weight of each meta-operation loop, assumed equal in this case. Substituting the resulting impact capability into Eq. (9) and Eq. (10) yields the performance estimate derived purely from operation-loop enumeration. Using the same damaged network and the same recovery sequence, we plotted the performance trajectories of the two metrics over time, as shown in Fig. 16.

In each subfigure of Fig. 16, the two recovery trajectories exhibit markedly different sensitivities. In some time steps, the number of operation loops increases substantially while no task chains are activated, as observed in the early stage of configuration cases 5. In other steps, although only a few operation loops are restored, repairing a key bottleneck link can immediately enable new task chains and thus trigger a pronounced jump in the TDMN-based metric, as observed in the mid stage of configuration cases 5 and the early stage of configuration cases 9. These phenomena indicate that task-chain formation is not linearly additive with respect to loop counts, but instead depends on coordinated improvements across multiple capability types. Moreover, they demonstrate that the TDMN-based metric can effectively filter out redundant structural enhancements and highlight recovery actions that genuinely improve USoS performance, making it a more reliable indicator of resilience.

4.3. Comparison of optimization methods

4.3.1. Algorithm parameter settings

Before validating the algorithm, it is first necessary to determine the values of parameter. The BMRMFO contains three key parameters: (1) population size (*PS*); (2) the number of iterations (*NI*); (3) depth of local enhancement (*LS*). To investigate the impact of parameter settings on the performance of BMRMFO, each parameter was assigned three levels, as shown in Table 6. In the parameter combination study, the orthogonal experimental design method was applied. An orthogonal array $L_9(3^3)$ containing nine parameter combinations was selected, and the specific settings are listed in Table 7.

Each combination was executed 12 times, and the results were integrated into the corresponding Pareto archive $E_i(i = 1, 2, ..., 9)$. All these archives were then merged into the final non-dominated solution set *FE*, where represents the total archive. The contribution of each parameter combination to was calculated as $CON_I = |E_i'|/|FE|$, where $E_i' = E_i \cap FE$. The contribution of was used as the response value *RV* for that combination.

Fig. 17 presents the experimental results, indicating that the optimal settings are: the second level for population size, the third level for number of iterations, and the second level for depth of local enhancement. Therefore, in the subsequent experiments for each algorithm, we set $NI = 150$, $PS = 20$ and $LS = 60$ respectively.

Table 6
Parameters for different levels.

Parameter	Factor level		
	1	2	3
NI	80	100	150
PS	15	20	25
LS	30	60	90

4.3.2. Comparison of algorithmic solution performance

The *C*-metric was chosen as the criterion to compare the dominant relationships between the solutions in the two Pareto files by two different algorithms. The definition of the *C* metric can be found in [32].

(a) Effectiveness Verification

To evaluate the effectiveness of the proposed algorithm, we constructed two test algorithms by modifying certain components: (1) BMRMFO-R abandons the use of the BMR initialization method and instead generates the initial solutions randomly; (2) BMRMFO-N disables the individual optimization process. In addition, NSGA-II and MOPSO also use random initialization strategies. These test algorithms were applied to various instances under configuration cases 5, considering different disturbance intensities and resource availabilities. The specific scenario settings are detailed in Table 8. Each scenario was run

Table 7
Taguchi orthogonal experiment parameter settings.

ID	*NI*	*PS*	*LS*	*RV*
1	1	1	1	0
2	1	2	3	0.167
3	1	3	2	0.083
4	2	1	3	0
5	2	2	2	0.083
6	2	3	1	0
7	3	1	2	0.417
8	3	2	1	0.25
9	3	3	3	0

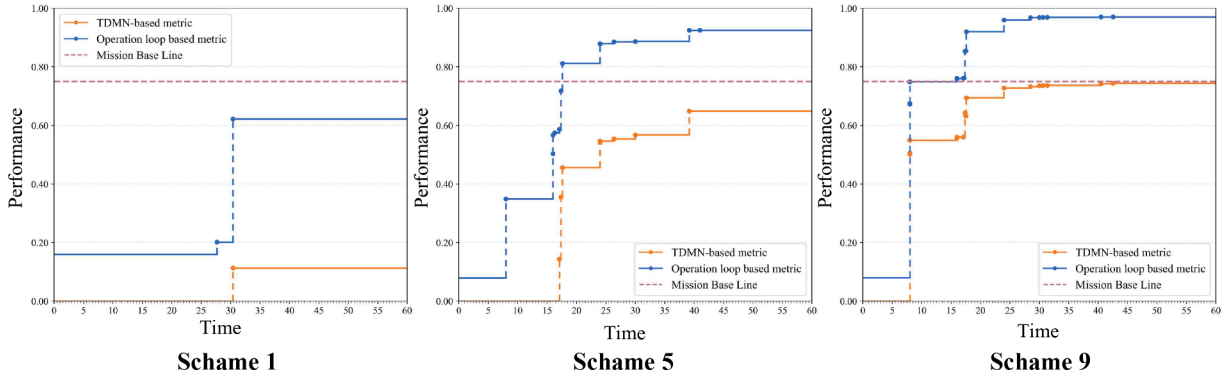


Fig. 16. Comparison of USoS recovery processes and evaluation metrics.

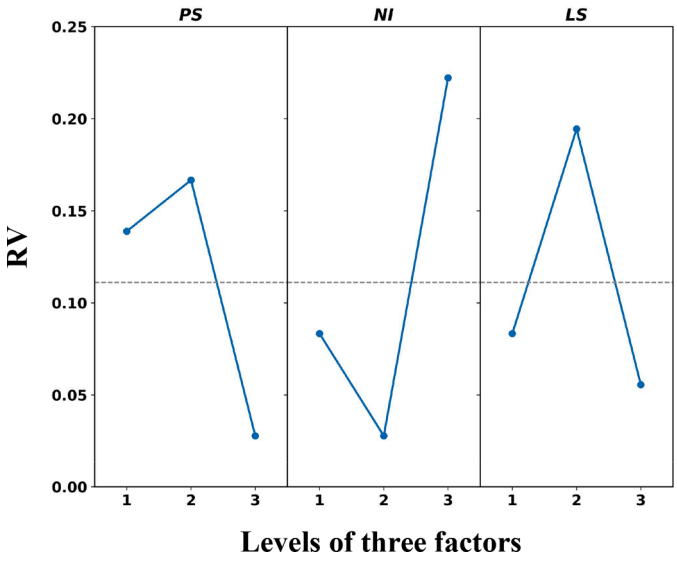


Fig. 17. Plots for main effects of RV.

independently five times, and the *C*-metric was used to evaluate each algorithm's performance.

Fig. 18 shows that in nearly all instances, the value of *C*(*BMRMFO/BMRMFO-R*) is greater than *C*(*BMRMFO-R/BMRMFO*), indicating the effectiveness of the proposed BMR-based initialization method. Similarly, *C*(*BMRMFO/BMRMFO-N*) is greater than *C*(*BMRMFO-N/BMRMFO*), demonstrating that the local reinforcement contributes positively to the performance of BMRMFO. These advantages persist across instances with varying solution space complexity and computational difficulty. In comparative experiments with NSGA-II and MOPSO, both *C*(*BMRMFO/NSGA-II*) and *C*(*BMRMFO/MOPSO*) are greater than their counterparts in the vast majority of instances, further confirming that the BMRMFO algorithm outperforms NSGA-II and MOPSO.

In addition, we compared the performance of the five algorithms under different numbers of entities in $d_c = 30$, $d_p = 20$ and $n_k = 4$. The scenario configurations are listed in Table 9. Each algorithm was run independently five times per scenario, and the resulting *C*-metric line plots are presented in Fig. 19.

The results in Fig. 19 show that BMRMFO consistently outperforms BMRMFO-R, BMRMFO-N, NSGA-II, and MOPSO under all conditions. It is worth noting that BMRMFO-R temporarily outperforms BMRMFO as the network scale increases. This occurs because the initial solutions provided by BMR may occasionally converge to local optima under specific configurations and damage scenarios, causing their performance to approach that of solutions generated through random initialization. It can be observed that as the problem scale increases, the superiority of BMRMFO becomes increasingly evident, demonstrating its capability in solving large-scale problems.

Based on the results of the two experiments above, the Wilcoxon signed-rank tests ($\alpha_s = 0.05$, one-sided) were performed based on the *C*-metric values under all intensities. As shown in Table 10, BMRMFO achieves much larger R+ values and near-zero R- values across all comparisons, with p-values < 0.001, indicating statistically significant superiority. Specifically, BMRMFO significantly outperforms its reduced variants as well as the baseline algorithms NSGA-II and MOPSO. These results confirm that the proposed hybrid mechanism and recovery-quality regulation strategy effectively enhance both convergence and diversity of the obtained Pareto fronts.

Finally, we calculated the Pareto fronts of each algorithm under the settings of $n = 50$, $d_c = 30$, $d_p = 20$ and $n_k = 4$. As shown in Fig. 20, where both objectives are minimized and better non-dominated solutions lie closer to the lower-left corner, BMRMFO produces a better distributed and higher quality Pareto set than BMRMFO-R, BMRMFO-N, NSGA-II, and MOPSO. Although BMRMFO-N achieves smaller cumulative performance loss resilience in some cases, its total recovery time cost is substantially higher, so it does not attain Pareto superiority.

Table 8
C-metric comparison of all algorithms under varying disruption intensities and support resources.
|(A = *C*(BMRMFO/BMRMFO-R), A′ = *C*(BMRMFO-R/BMRMFO), B = *C*(BMRMFO/BMRMFO-N), B′ = *C*(BMRMFO-N/BMRMFO), C = *C*(BMRMFO/NSGA-II), C′ = *C*(NSGA-II/BMRMFO), D = *C*(BMRMFO/MOPSO), D′ = *C*(MOPSO/BMRMF).

Instance	A	A'	B	B'	C	C'	D	D'
30 × 2	**0.75**	0	**0.5**	0	**0.22**	0	**0.5**	0.33
30 × 4	**0.6**	0.08	**0.75**	0.31	**0.61**	0.31	**0.57**	0.38
30 × 6	**0.33**	0.29	**0.62**	0.14	**0.96**	0	0.09	**0.71**
50 × 2	**0.5**	0.25	**1**	0	**0.8**	0	**0.8**	0.25
50 × 4	**0.6**	0.29	**0.8**	0.19	**0.58**	0.14	**0.88**	0.14
50 × 6	**0.75**	0	**1**	0	0	**0.22**	**1**	0
70 × 2	**0.5**	0	**1**	0	**0.92**	0	**0.73**	0
70 × 4	**0.62**	0.25	**1**	0	**1**	0	**0.71**	0.25
70 × 6	**0.57**	0.33	**0.83**	0	**1**	0	**0.4**	0.17
Average	**0.58**	0.17	**0.77**	0.08	**0.68**	0.07	**0.54**	0.30

(b) Relay-priority examination

To further examine whether different optimization algorithms exhibit the desired relay-centric recovery behavior, we analyze the recovery sequences generated under all intensities. For each recovered entity, we record its hop distance to the nearest relay entity as well as the recovery-quality level assigned by the algorithm. Based on these sequences, we compute sequential earliness E_e and resource priority E_r to quantify the early recovery preference toward relay entities and their close neighbors, and the two indicators are individually normalized as $\widetilde{E}_e$, $\widetilde{E}_r$. And then averaged to obtain a unified metric $\overline{E}$ for comparing the algorithms.

For each occurrence of entities whose distance to a relay entity belongs to the set 0, 1 and 2. We compute its normalized position to reflect the higher importance of closer-to-relay nodes, we assign distance-dependent weights: $\omega_r(0) = 1$, $\omega_r(1) = 0.2$,$\omega_r(2) = 0.1$. The weighted earliness of a sequence is then defined as Eqs. (31) and (32):

$$E_e = \frac{\sum_{i\in\Omega}\left(st_i \cdot w_r(d_r) \Big/ \sum_i st_i\right)}{\sum_{i\in\Omega} w_r(d_r)} \tag{31}$$

$$E_r = \frac{\sum_{i\in\Omega}\left(\sum_q (z_{i,q}\cdot q)\cdot w_r(d_i) \Big/ \sum_{i,q}(z_{i,q}\cdot q)\right)}{\sum_{i\in\Omega} w_r(d_i)} \tag{32}$$

where Ω is the index set of relay-related entity in the sequence. st_i represents the position of entity i within that sequence.

Table 11 indicates that BMRMFO achieves the best overall performance, demonstrating its ability to restore relay entities and their neighboring entities earlier and to allocate higher-quality recovery resources to them, consistent with its design mechanisms. Although MOPSO exhibits a certain level of relay preference, it remains noticeably weaker than BMRMFO. BMRMFO-N deteriorates on both metrics due to the absence of local reinforcement. Neither NSGA-II nor BMRMFO-R shows stable or significant relay-priority behavior, with BMRMFO-R exhibiting more scattered and unstable patterns because of its lack of structured search guidance.

4.3.3. Comparison of recovery strategy superiority

To demonstrate the superiority of BMRMFO over traditional algorithms, we selected five importance-based recovery strategy decision

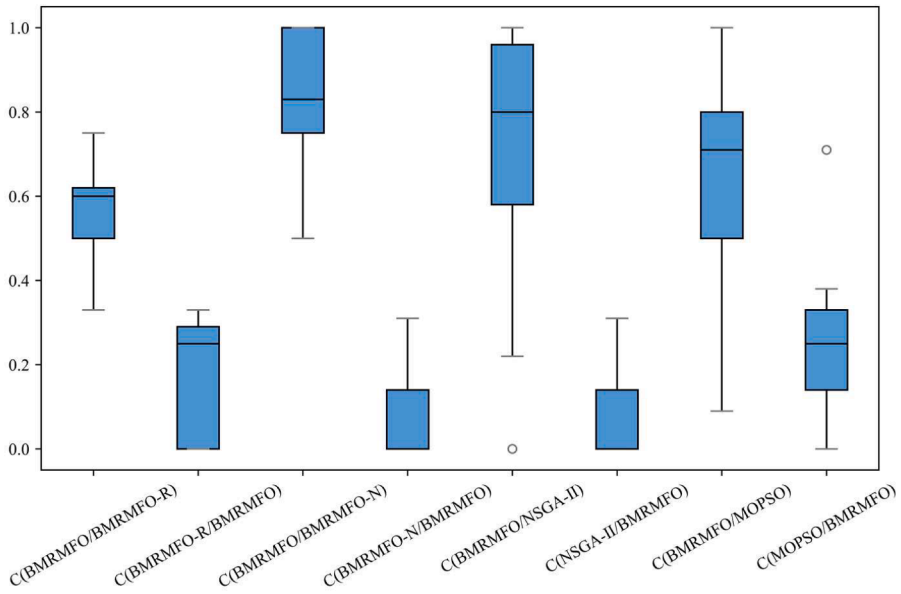


Fig. 18. The boxplots of C metrics of BMRMFO and other algorithms.

Table 9
C-metric comparison of all algorithms under different USoS configuration cases. (A = *C*(BMRMFO/BMRMFO-R), A′ = *C*(BMRMFO-R/BMRMFO), B = *C*(*BMRMFO/BMRMFO-N*), B′ = *C*(BMRMFO-N/BMRMFO), C = *C*(BMRMFO/NSGA-II), C′ = *C*(NSGA-II/BMRMFO), D = *C*(BMRMFO/MOPSO), D′ = *C*(MOPSO/BMRMF),.

Instance	A	A'	B	B'	C	C'	D	D'
30	**0.57**	0	**0.86**	0	**0.67**	0.17	**0.33**	0.17
35	**0.71**	0.14	**1**	0	**1**	0	**0.45**	0.36
40	**0.50**	0.17	**1**	0	**1**	0	**0.86**	0
45	**1**	0	**1**	0	**1**	0	**0.6**	0.4
50	**0.6**	0.29	**0.8**	0.19	**0.58**	0.14	**0.88**	0.14
55	0.33	**0.43**	**1**	0	**1**	0	**1**	0
60	**0.22**	0	**1**	0	**0.33**	0	**1**	0
65	**0.67**	0	**1**	0	**0.22**	0	**0.83**	0
70	**1**	0	**1**	0	**0.93**	0	**0.57**	0
Average	**0.62**	0.11	**0.92**	0.03	**0.75**	0.03	**0.72**	0.12

algorithms [2] and a random recovery strategy decision algorithm for comparison with the proposed optimization algorithm. These algorithms are described as follows:

- Degree centrality-based recovery strategy (DCRS). The damaged entities are restored in descending order of their degree centrality values.
- Betweenness centrality-based recovery strategy (BCRS). The damaged entities are recovered in descending order of their betweenness centrality values.
- Closeness centrality-based recovery strategy (CCRS). The damaged entities are recovered in descending order based on their closeness centrality values.
- Eigenvector centrality-based recovery strategy (ECRS). The damaged entities are recovered in descending order of their eigenvector centrality values.
- Clustering Coefficient-based Recovery Strategy (CLUST). The damaged entities are recovered in descending order based on the clustering coefficient, which reflects the degree of interconnectedness among their neighboring entities.
- Random Recovery Strategy (RRS). The damaged entities are recovered in descending order based on the random strategy.

We primarily evaluated the superiority of BMRMFO over baseline algorithms under different disruption intensities and support site configurations, based on the USoS configuration described in configuration cases 5. To comprehensively assess the performance of BMRMFO, six instances were constructed in different total damage ratios and support site configurations, representing Recovery After Damage scenarios of different complexities. The detailed settings are presented in Table 12.

Recovery was performed using six baseline strategies and the proposed BMRMFO algorithm. Table 13 presents the objective values obtained by each method in the six test instances, where BMRMFO selects the solution with the minimum $\mathscr{R}_{SoS}$. Compared to traditional centrality-based strategies, BMRMFO achieved an average improvement of 27.62 % in $\mathscr{R}_{SoS}$, and 22.61 % in TF_{SoS}. Although selecting the solution with the minimum value of $\mathscr{R}_{SoS}$ results in BMRMFO being slightly inferior to CCRS on TF_{SoS} in instance 1, the overall results still provide strong evidence of BMRMFO's superior performance across diverse cases, particularly in large-scale Recovery After Damaged scenarios.

The solutions generated by each method were decoded into corresponding recovery strategies, and their resilience trajectories were evaluated through simulation, as illustrated in Fig. 21. Compared with all baseline algorithms, BMRMFO yields resilience curves with a substantially larger enclosed area and achieves a higher post-recovery performance level for the USoS. This advantage is particularly evident in Instances 1 and 4, where most baseline strategies fail to restore system performance to the required task baseline, whereas BMRMFO consistently ensures that the final performance exceeds this threshold.

4.4. Discussion

This study validates, through multi-instance and multi-algorithm comparative experiments, that the proposed evaluation–optimization framework can systematically enhance the resilience of relay-enabled USoS. At the evaluation level, the results indicate that tasks, as higher-level abstractions of system capabilities, imply that localized improvements in individual capabilities do not necessarily translate into mission-level performance gains; instead, meaningful performance improvements can be achieved only when the constituent capabilities within a task chain evolve in a coordinated manner.

In the summary, the results indicate that BMRMFO consistently yields more effective recovery strategies under complex and heterogeneous conditions. Compared with conventional methods, BMRMFO enables more efficient performance restoration while maintaining balanced trade-offs among competing recovery objectives,

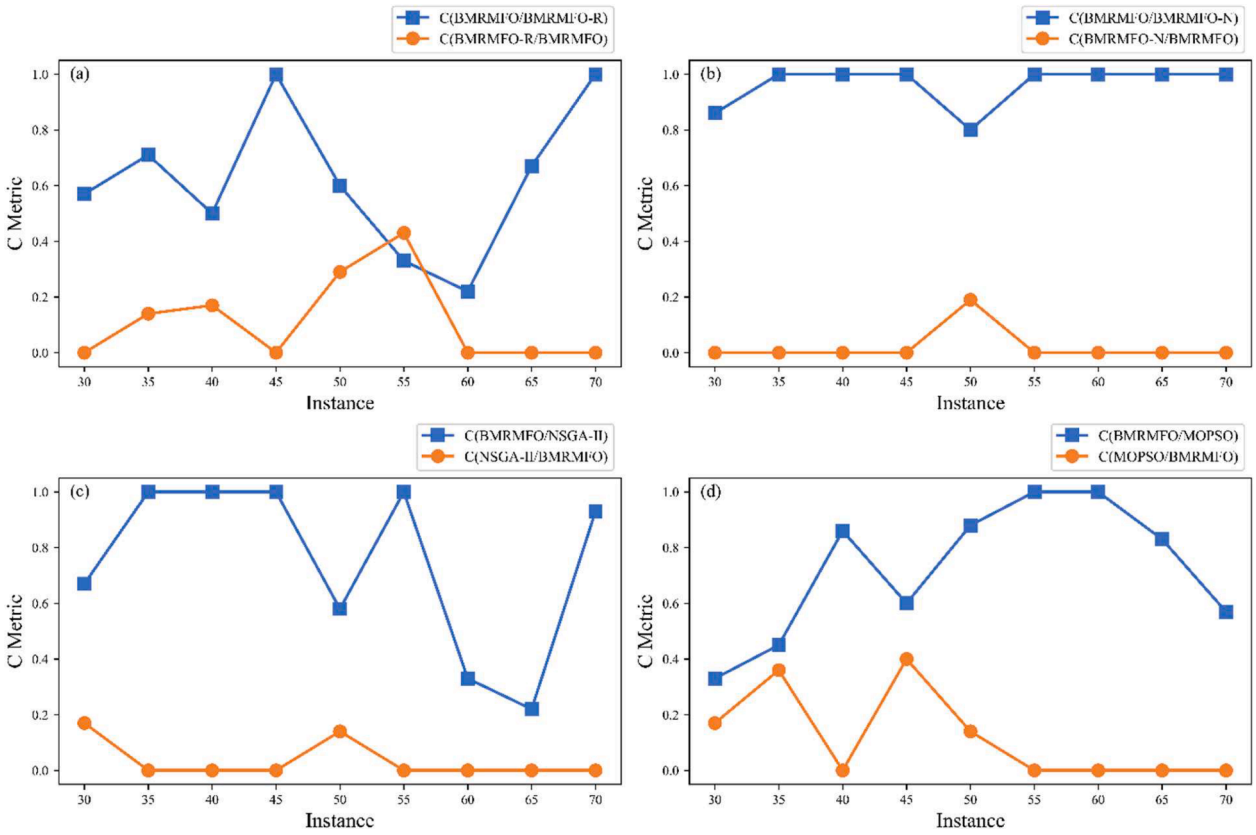


Fig. 19. C-metric comparison of TSKCA and other algorithms under different instances.

Table 10
Wilcoxon's test of C-metric of all algorithms.

BMRMFO *vs*	R+	R-	Z	*p*-value	$\alpha_s = 0.05$
BMRMFO-R	169	2	3.636	0.0000	**Yes**
BMRMFO-N	171	0	3.724	0.0000	**Yes**
NSGA-II	169	2	3.636	0.0000	**Yes**
MOPSO	161	10	3.288	0.0002	**Yes**

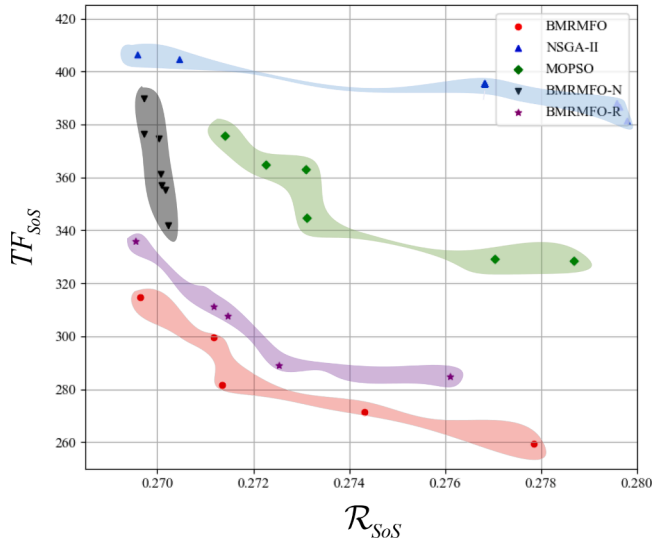


Fig. 20. The obtained Pareto frontiers.

Table 11
Relay-priority evaluation results.

Algorithm	E_e	E_r	$\widetilde{E}_e$	$\widetilde{E}_r$	$\overline{E}$
BMRMFO	0.855	0.483	1.000	0.515	0.758
BMRMFO-R	0.832	0.480	0.800	0.593	0.696
BMRMFO-N	0.805	0.501	0.540	0.000	0.270
NSGA-II	0.746	0.465	0.000	1.000	0.500
MOPSO	0.849	0.490	0.950	0.323	0.635

Table 12
The specific settings of instances.

Instance	n	n_k	d_r
1	30	2	50
2	50	4	50
3	70	6	50
4	30	2	70
5	50	4	70
6	70	6	70

demonstrating stable behavior in relay-enabled USoS. Notably, the recovery process exhibits an implicit tendency toward relay-prioritized restoration: even without explicitly enforced relay-first rules, BMRMFO naturally favors relay entities and their neighborhood which suggests that the integration of TDMN-based evaluation with BMRMFO optimization allows the algorithm to internalize the structural and functional role of relays in sustaining task-chain integrity, thereby directing recovery actions toward mission-critical components and promoting resilience enhancement. However, the experiments rely on static communication assumptions and deterministic delays, without accounting for dynamic channel fading or stochastic interference. Future work may extend TDMN to stochastic and dynamic

Table 13
Comparison of $\mathscr{R}_{SoS}$ and TF_{SoS} for all algorithms.

Instance	$\mathscr{R}_{SoS}$							TF_{SoS}						
	DCRS	BCRS	CCRS	ECRS	CLUST	RRS	BMRMFO	DCRS	BCRS	CCRS	ECRS	CLUST	RRS	BMRMFO
1	0.65	0.58	0.68	0.77	0.68	0.59	**0.48**	281	268	**251**	269	258	294	257
2	0.30	0.39	0.32	0.35	0.30	0.40	**0.26**	495	521	530	516	519	595	**367**
3	0.40	0.54	0.40	0.40	0.45	0.51	**0.32**	749	754	759	740	756	881	**616**
4	0.68	0.63	0.64	0.72	0.63	0.73	**0.50**	407	387	397	417	416	423	**312**
5	0.42	0.52	0.42	0.46	0.49	0.78	**0.35**	714	740	746	712	751	865	**449**
6	0.50	0.54	0.47	0.51	0.49	0.65	**0.38**	1047	1012	1000	1056	1054	1137	**885**

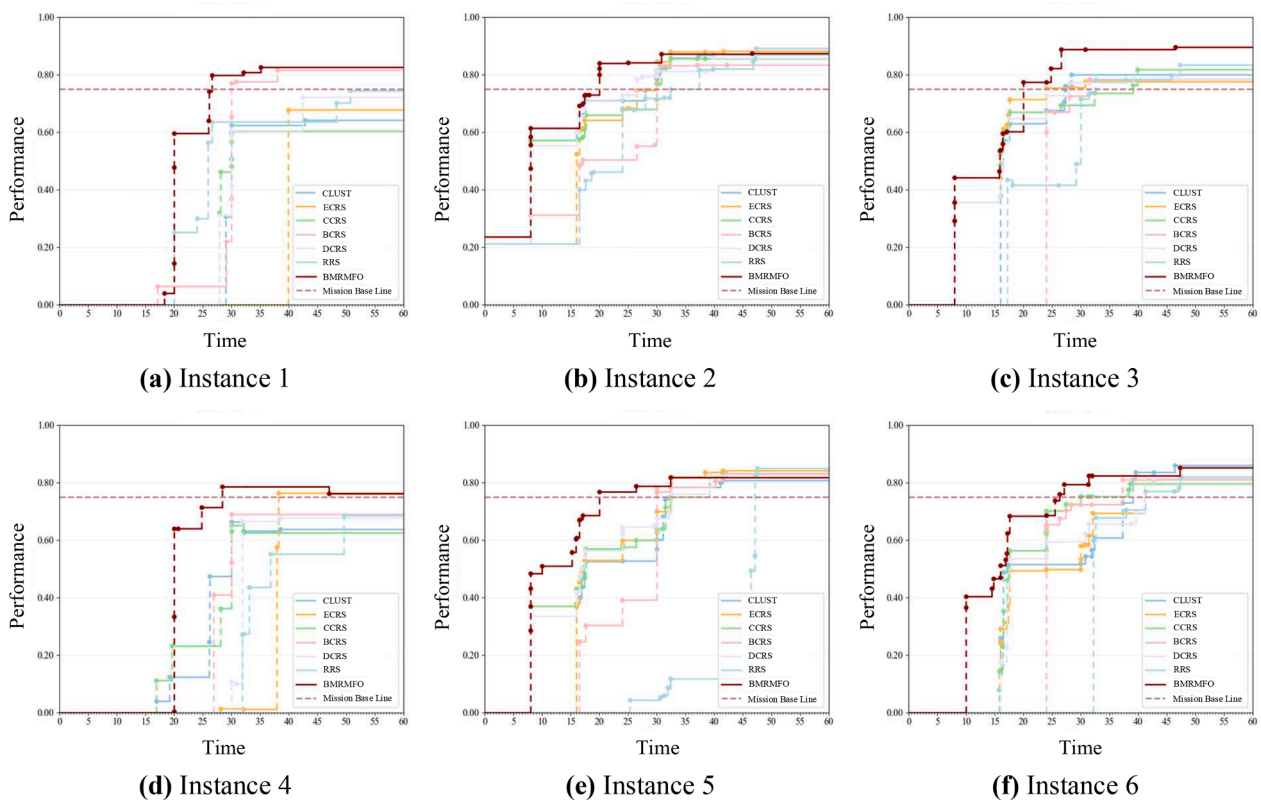


Fig. 21. Comparison of BMRMFO and baseline methods under varying disruption intensities and support resources.

communication models, enabling adaptive resilience assessment and recovery optimization under time-varying channel conditions.

5. Conclusion

The introduction of relay communication systems improves communication coverage and coordination flexibility in USoS, while increasing structural coupling and vulnerability. To address this challenge, this study proposes a resilience modeling and recovery optimization approach which explicitly accounts for relay-based communication mechanisms. First, a TDMN is constructed to quantitatively characterize the performance gains induced by relays and to establish a task-centric evaluation structure. Second, the BMRMFO algorithm is developed to jointly minimize cumulative performance loss resilience and total recovery time cost by prioritizing relay-dependent capability pathways and task chains. Extensive case studies are conducted to verify the effectiveness and superiority of the proposed TDMN and BMRMFO, and to analyze the structural and functional characteristics of USoS after the introduction of relay communication.

Future work may further refine the communication mechanisms of relay systems and extend the performance indicators to include packet loss rate, bandwidth, and other communication-related metrics, thereby improving the practical applicability of the proposed approach. In addition, recovery scenarios may be expanded toward Immediate Recovery Scenarios settings with a broader range of available recovery measures.

CRediT authorship contribution statement

Xu Wang: Writing – original draft, Visualization, Software, Methodology, Conceptualization. **Hengte Du:** Writing – review & editing, Validation, Methodology. **Yuheng Dang:** Methodology, Conceptualization. **Xing Pan:** Writing – review & editing, 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.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 72071011.

Appendix

1) Indices

l	The index of task types
p	The index of mate-task chain types
a	The index of targets
i	The index of entities
j	The index of recovery measures
k	The index of support sites
q	The index of recovery quality levels

2) Parameters

S	The symbol of sensor entity
D	The symbol of decider entity
I	The symbol of influencer entity
T	The symbol of target entity
$\mathscr{N}$	The Task-driven Multilayer Network
$\mathscr{S}$	The Structure Layer network
$\mathscr{C}$	The Capability Layer network
$\mathscr{M}$	The Mission Layer network
V^{lt}	The node set of each network in TDMN, $lt \in \{\mathscr{S}, \mathscr{C}, \mathscr{M}\}$
E^{lt}	The edge set of each network in TDMN, $lt \in \{\mathscr{S}, \mathscr{C}, \mathscr{M}\}$
V^{et}	The sets of each entity type, $et \in \{S, D, I, T\}$
E^{C}	The sets of functional interaction edges
E^{F}	The sets of interaction relationships edges
$W^{\mathscr{C}}$	A weight function of Capability Layer network
$V^{R:S,D,I}$	The sets of relay entities
$V^{N:S,D,I}$	The sets of non-relay entities
N_R	The number of relay entities
$\mathscr{P}_p$	The meta-task chain p
m_p	The number of task types of meta-task chain p
n_p	The number of meta-task chain
L_l	The task type l
v_i	The entity i
$D(v_i)$	The functional integrity level of entity i
$c_{y,l,a}$	The operation loop y of type l associated with target a
$C(c_{y,l,a})$	The performance of the operation loop y
$L(c_{y,l,a})$	The total number of communication hops along all edges in operation loop y
$L_{min,l}$	The minimal number of hops required to complete the meta-operation loop l
$F(c_{y,l,a})$	The overall capability of operation loop y
$n_{l,a}$	The total number of tasks type l to target a
$\overline{E}_{l,a}$	Average task-support level for task type l to target a
TP_a	The list of required meta-task chain of target a
$Ln_{a,p}$	The number of meta-task chain p associated with the target a
$\omega_{p,a}$	The importance weight of meta-task chain p for target a
R_a	The Influence Effect on target a
λ_a	The normalized importance weight of target a
A	The total number of target entities
α	The marginal decay factor of influence effect
ψ_{SoS}	The overall USoS performance
P	The normalized performance of USoS
T_e	The recovery termination time
T_t	The recovery completion time
t_{max}	The maximum recovery time cost allowed for USoS
$F_{spt,j}$	Unit time operation cost of recovery measure type j
F_{otc}	The operation time of recovery strategy
F_{mtc}	The mobilization time of recovery strategy
K_k	Support sites k
$d_{i,k}$	The distance from entity v_i to each support site k
Qe_q	Recovery-effect coefficient
Qt_q	Recovery-time coefficient
hop_{i_1,i_2}	The minimum number of hops required to establish communication between entity i1 and i2
n	Total number of entities
n_k	Total number of support sites
n_q	Total number of support sites recover quality levels,
D_i^*	The functional integrity level after recovery of entity i
τ	Capability degradation coefficient
v_s	Mobilization speed of maintenance vehicles
v_m	Mobilization speed of the entity

(*continued on next page*)

(*continued*)

HOP_{max}	Relay-based communication hop count upper bound
TH_{max}	Total loop length total loop length
r_c	The communication range of each entity
r_s	The observation range of sensors
r_l	The strike range of influencers
d_r	Total damaged ratio
d_c	Complete damage ratio
d_p	Partial damage ratio
TR_k	The storage capacity of support site k
P_{BL}	Mission baseline
RB	The number of redundant backups
$\mathscr{R}_{SoS}$	The cumulative performance loss resilience
TF_{SoS}	Total recovery time cost

3) Decision variables

$x_{i,j,k}$	Binary variable equal to 1 if recovery measure j is performed on entity i at support site k; otherwise, 0.
$z_{i,q}$	Binary variable equal to 1 if entity i selects recovery quality level q; otherwise, 0.

Data availability

Data will be made available on request.

References

[1] Chen J, Sun J, Wang G. From unmanned systems to autonomous intelligent systems. Engineering 2022;12:16–9. https://doi.org/10.1016/j.eng.2021.10.007.

[2] Xu R, Ning G, Liu J, Li M, Li J, Yang K, et al. CSoS-STRE: a combat system-of-system space-time resilience enhancement framework. Front Eng Manag 2025. https://doi.org/10.1007/s42524-025-4179-y.

[3] Wang K, Zhao T, Yuan Y, Hao Z, Chen Z, Dui H. A new multi-layer performance analysis of unmanned system-of-systems within IoT. Reliab Eng Syst Saf 2025;259:110953. https://doi.org/10.1016/j.ress.2025.110953.

[4] El Debeiki M, Al-Rubaye S, Perrusquía A, Conrad C, Flores-Campos JA. An advanced path planning and UAV relay system: enhancing connectivity in rural environments. Future Internet 2024;16:89. https://doi.org/10.3390/fi16030089.

[5] Baek H, Lim J. Design of future UAV-relay tactical data link for reliable UAV control and situational awareness. IEEE Commun Mag 2018;56:144–50. https://doi.org/10.1109/mcom.2018.1700259.

[6] Khan N, Ahmad A, Wakeel A, Kaleem Z, Rashid B, Khalid W. Efficient UAVs deployment and resource allocation in UAV-relay assisted public safety networks for video transmission. IEEE Access 2024;12:4561–74. https://doi.org/10.1109/access.2024.3350138.

[7] Liu X, Yu Y, Li F, Durrani TS. Throughput maximization for RIS-UAV relaying communications. IEEE Trans Intell Transp Syst 2022;23:19569–74. https://doi.org/10.1109/tits.2022.3161698.

[8] Xing L. Cascading failures in Internet of Things: review and perspectives on reliability and resilience. IEEE Internet Things J 2021;8:44–64. https://doi.org/10.1109/JIOT.2020.3018687.

[9] Fu X, Yang Y. Modeling and analyzing cascading failures for Internet of Things. Inf Sci 2021;545:753–70. https://doi.org/10.1016/j.ins.2020.09.054.

[10] Fu X, Li Q, Li W. Modeling and analysis of industrial IoT reliability to cascade failures: an information-service coupling perspective. Reliab Eng Syst Saf 2023;239:109517. https://doi.org/10.1016/j.ress.2023.109517.

[11] Liu T, Bai G, Tao J, Zhang Y-A, Fang Y. A multistate network approach for resilience analysis of UAV swarm considering information exchange capacity. Reliab Eng Syst Saf 2024;241:109606. https://doi.org/10.1016/j.ress.2023.109606.

[12] Feng Q, Liu M, Dui H, Ren Y, Sun B, Yang D, et al. Importance measure-based phased mission reliability and UAV number optimization for swarm. Reliab Eng Syst Saf 2022;223:108478. https://doi.org/10.1016/j.ress.2022.108478.

[13] Liu X, Li D, Ma M, Szymanski BK, Stanley HE, Gao J. Network resilience. Phys Rep 2022;971:1–108. https://doi.org/10.1016/j.physrep.2022.04.002.

[14] Eldosouky A, Saad W, Mandayam N. Resilient critical infrastructure: Bayesian network analysis and contract-based optimization. Reliab Eng Syst Saf 2021;205:107243. https://doi.org/10.1016/j.ress.2020.107243.

[15] Pan X, Dang Y, Wang H, Hong D, Li Y, Deng H. Resilience model and recovery strategy of transportation network based on travel OD-grid analysis. Reliab Eng Syst Saf 2022;223:108483. https://doi.org/10.1016/j.ress.2022.108483.

[16] Kristianto Y, Gunasekaran A, Helo P, Hao Y. A model of resilient supply chain network design: a two-stage programming with fuzzy shortest path. Expert Syst Appl 2014;41:39–49. https://doi.org/10.1016/j.eswa.2013.07.009.

[17] Wu C, Deng H, Wu H, Tu C. Enhancing resilience of unmanned autonomous swarms through game theory-based cooperative reconfiguration. Reliab Eng Syst Saf 2025;260:110951. https://doi.org/10.1016/j.ress.2025.110951.

[18] Sun Q, Li H, Wang Y, Zhang Y. Multi-swarm-based cooperative reconfiguration model for resilient unmanned weapon system-of-systems. Reliab Eng Syst Saf 2022;222:108426. https://doi.org/10.1016/j.ress.2022.108426.

[19] Mou Z, Gao F, Liu J, Wu Q. Resilient UAV swarm communications with graph convolutional neural network. IEEE J Sel Areas Commun 2022;40:393–411. https://doi.org/10.1109/jsac.2021.3126047.

[20] Zhu X, Zhu X, Yan R, Peng R. Optimal routing, aborting and hitting strategies of UAVs executing hitting the targets considering the defense range of targets. Reliab Eng Syst Saf 2021;215:107811. https://doi.org/10.1016/j.ress.2021.107811.

[21] Xu W, Jaimoukha IM, Teng F. Robust moving target defence against false data injection attacks in power grids. IEEE Trans Inf Forensics Secur 2023;18:29–40. https://doi.org/10.1109/TIFS.2022.3210864.

[22] Li J, Zhao D, Jiang J, Yang K, Chen Y. Capability oriented equipment contribution analysis in temporal combat networks. IEEE Trans Syst Man Cybern Syst 2021;51:696–704. https://doi.org/10.1109/tsmc.2018.2882782.

[23] Li J, Jiang J, Yang K, Chen Y. Research on functional robustness of heterogeneous combat networks. IEEE Syst J 2019;13:1487–95. https://doi.org/10.1109/jsyst.2018.2828779.

[24] Al-Kofahi O, Kamal A. Survivability strategies in multihop wireless networks. IEEE Wirel Commun 2010;17:71–80. https://doi.org/10.1109/MWC.2010.5601961.

[25] Wang L Z, Zhao X J, Zhang Y, Wang X H, Ma T L, Gao X. Unmanned aerial vehicle swarm mission reliability modeling and evaluation method oriented to systematic and networked mission. Chin J Aeronaut 2021;34(2):466–78. https://doi.org/10.1016/j.cja.2020.02.026.

[26] Kong L, Wang L, Cao Z, Wang X. Resilience evaluation of UAV swarm considering resource supplementation. Reliab Eng Syst Saf 2024;241:109673. https://doi.org/10.1016/j.ress.2023.109673.

[27] Fu X, Yao H, Yang Y. Cascading failures in wireless sensor networks with load redistribution of links and nodes. Ad Hoc Netw 2019;93:101900. https://doi.org/10.1016/j.adhoc.2019.101900.

[28] Lv H, Wu Z, Zhang X, Jiang B, Gao Q. Cascading failure analysis of hierarchical industrial wireless sensor networks under the impact of data overload. Machines 2022;10:380. https://doi.org/10.3390/machines10050380.

[29] Pan X. Resilience based importance measure analysis for SoS. JESS 2019;30:920. https://doi.org/10.21629/jsee.2019.05.10.

[30] Xu R, Liu J, Li J, Yang K, Long H, Lou Z. Enhancing the resilience of combat system-of-systems through a two-phase recovery framework. Reliab Eng Syst Saf 2026;265:111437. https://doi.org/10.1016/j.ress.2025.111437.

[31] Chen Z, Yin S, Li L, Cui W, Hong D. Resilience metric and dynamic assessment of unmanned system-of-systems considering cooperative reconfiguration strategies. IEEE Trans Reliab 2025;74:2942–54. https://doi.org/10.1109/tr.2024.3438810.

[32] Zhao F, He X, Wang L. A two-stage cooperative evolutionary algorithm with problem-specific knowledge for energy-efficient scheduling of No-wait flow-shop problem. IEEE Trans Cybern 2021;51:5291–303. https://doi.org/10.1109/tcyb.2020.3025662.

[33] Yu X, Du H, Zhou D, Wang Q, Lin G. Bi-objective carbon-efficient distributed flow-shop scheduling with multistep electricity pricing. Int J Prod Res 2024;62:6842–58. https://doi.org/10.1080/00207543.2024.2312209.

[34] Xu M, Ouyang M, Hong L, Mao Z, Xu X. Resilience-driven repair sequencing decision under uncertainty for critical infrastructure systems. Reliab Eng Syst Saf 2022;221:108378. https://doi.org/10.1016/j.ress.2022.108378.

[35] Lin M, Chen T, Chen H, Ren B, Zhang M. When architecture meets AI: a deep reinforcement learning approach for system of systems design. Adv Eng Inform 2023;56:101965. https://doi.org/10.1016/j.aei.2023.101965.
[36] Huang Y, Luo A, Chen T, Zhang M, Ren B, Song Y. When architecture meets RL+EA: a hybrid intelligent optimization approach for selecting combat system-of-systems architecture. Adv Eng Inform 2023;58:102209. https://doi.org/10.1016/j.aei.2023.102209.
[37] Hu T, Zong Y, Lu N, Jiang B. Toward the resilience of UAV swarms with percolation theory under attacks. Reliab Eng Syst Saf 2025;254:110608. https://doi.org/10.1016/j.ress.2024.110608.
[38] Xie L, Xu J, Zhang R. Throughput maximization for UAV-enabled wireless powered communication networks - invited paper. In: 2018 IEEE 87th Veh. Technol. Conf. VTC Spring. IEEE; 2018. p. 1–7. https://doi.org/10.1109/vtcspring.2018.8417659.
[39] Long H, Xiang W, Sun Y. Low-latency and high-reliability performance analysis of relay systems. IET Commun 2018;12:627–33. https://doi.org/10.1049/iet-com.2017.0680.
[40] Khandani A, Abounadi J, Modiano E, Zheng L. Reliability and route diversity in wireless networks. IEEE Trans Wirel Commun 2008;7:4772–6. https://doi.org/10.1109/T-WC.2008.070770.
[41] Wang L, Chen L, Yang Z, Yang K. Research on network robustness analysis method of combat system-of-systems considering collaborative reconfiguration. Reliab Eng Syst Saf 2026;265:111526. https://doi.org/10.1016/j.ress.2025.111526.
[42] Karim S, Capron L. Reconfiguration: adding, redeploying, recombining and divesting resources and business units. Strateg Manag J 2016;37. https://doi.org/10.1002/smj.2537.
[43] Hwang I, Kim S, Kim Y, Seah CE. A survey of fault detection, isolation, and reconfiguration methods. IEEE Trans Control Syst Technol 2010;18:636–53. https://doi.org/10.1109/tcst.2009.2026285.
[44] Sun Q, Li H, Zhong Y, Ren K, Zhang Y. Deep reinforcement learning-based resilience enhancement strategy of unmanned weapon system-of-systems under inevitable interferences. Reliab Eng Syst Saf 2024;242:109749. https://doi.org/10.1016/j.ress.2023.109749.
[45] Li J, Mosleh A, Kang R. Likelihood ratio gradient estimation for dynamic reliability applications. Reliab Eng Syst Saf 2011;96:1667–79. https://doi.org/10.1016/j.ress.2011.08.001.
[46] Bai G, Li Y, Fang Y, Zhang Y-A, Tao J. Network approach for resilience evaluation of a UAV swarm by considering communication limits. Reliab Eng Syst Saf 2020;193:106602. https://doi.org/10.1016/j.ress.2019.106602.
[47] Zhang P, Wu T, Cao R, Li Z, Xu J. UAV swarm resilience assessment considering load balancing. Front Phys 2022;10. https://doi.org/10.3389/fphy.2022.821321.
[48] Zhong Y, Li H, Zhuang X. A resilience-driven two-stage operational chain optimization model for unmanned weapon system-of-systems under limited resource environments. Eksploat Niezawodn – Maint Reliab 2024;26. https://doi.org/10.17531/ein/188198.
[49] Shelton CP, Koopman P. Improving system dependability with functional alternatives. In: Int. Conf. Dependable Syst. Netw. 2004. IEEE; 2004. p. 295–304. https://doi.org/10.1109/DSN.2004.1311899.
[50] Wenxing L, Muqing W, Min Z, Peizhe L, Tianze L. Hop count limitation analysis in wireless multi-hop networks. Int J Distrib Sens Netw 2017;13:155014771668360. https://doi.org/10.1177/1550147716683606.
[51] Vikas NSJ. Multi-objective moth flame optimization. In: 2016 Int. Conf. Adv. Comput. Commun. Inform. ICACCI. IEEE; 2016. p. 2470–6. https://doi.org/10.1109/icacci.2016.7732428.
[52] Zamani H, Nadimi-Shahraki MH, Mirjalili S, Soleimanian Gharehchopogh F, Oliva D. A critical review of moth-flame optimization algorithm and its variants: structural reviewing, performance evaluation, and statistical analysis. Arch Comput Methods Eng 2024;31:2177–225. https://doi.org/10.1007/s11831-023-10037-8.
[53] Pelusi D, Mascella R, Tallini L, Nayak J, Naik B, Deng Y. An Improved Moth-Flame Optimization algorithm with hybrid search phase. Knowl-Based Syst 2020;191:105277. https://doi.org/10.1016/j.knosys.2019.105277.
[54] Shehab M, Abualigah L, Al Hamad H, Alabool H, Alshinwan M, Khasawneh AM. Moth–flame optimization algorithm: variants and applications. Neural Comput Appl 2020;32:9859–84. https://doi.org/10.1007/s00521-019-04570-6.
[55] Li H, Liu J, Chen L, Bai J, Sun Y, Lu K. Chaos-enhanced moth-flame optimization algorithm for global optimization. JSEE 2019;30:1144–59. https://doi.org/10.21629/jsee.2019.06.10.