Front. Eng. Manag. 2025, 12(4): 1175–1195
https://doi.org/10.1007/s42524-025-4179-y

RESEARCH ARTICLE

Renjie XU, Guoyu NING, Jiahao LIU, Minghao LI, Jichao LI, Kewei YANG, Zhiyuan LOU

CSoS-STRE: A combat system-of-system space-time resilience enhancement framework

Abstract A combat system-of-systems (CSoS) is a network of independent entities that interact to provide overall operational capabilities. Enhancing the resilience of CSoS is garnering increasing attention due to its practical value in optimizing network architectures, improving network security and refining operational planning. Accordingly, we present a unified framework called CSoS space-time resilience enhancement (CSoS-STRE) to enhance the resilience of CSoS. Specifically, we develop a spatial combat network model and a space-time resilience optimization model that captures the complex spatial relationships between entities and reformulates the resilience enhancement problem as a linear optimization model with spatial features. Moreover, we extend the model to include obstacles. Next, a resilience-oriented recovery optimization method based on the improved non-dominated sorting genetic algorithm II (R-INSGA) is proposed to determine the optimal recovery sequence for the damaged entities. This method incorporates spatial features while providing the optimal travel paths for multiple recovery teams. Finally, the feasibility, effectiveness, and superiority of the CSoS-STRE are demonstrated through a case study, providing valuable insights for guiding recovery and developing more resilient CSoS.

Keywords combat system-of-systems, space-time resilience enhancement, resilience optimization model, recovery optimization method, no-obstacles and obstacles

Received Sep. 20, 2024; revised Dec. 18, 2024; accepted Jan. 1, 2025

Renjie XU, Guoyu NING, Jiahao LIU, Minghao LI, Jichao LI (✉), Kewei YANG (✉)
College of Systems Engineering, National University of Defense Technology, Changsha 410073, China
E-mails: ljcnudt@hotmail.com; kayyang27@nudt.edu.cn

Zhiyuan LOU
School of Management, Technical University of Munich, Heilbronn 74076, Germany

This study was supported in part by the National Natural Science Foundation of China (Grant Nos. 72371244, 72301286, 72231011, and 72431011); Postgraduate Scientific Research Innovation Project of Hunan Province, China (No. CX20240146).

1 Introduction

Modern warfare has transitioned from the paradigm of isolated combat forces to system-to-system confrontations due to advancements in combat technologies and application concepts. The US Navy and the Defense Advanced Research Projects Agency (DARPA) introduced the groundbreaking concepts of network-centric warfare (Sinex et al., 2000) and mosaic warfare (Deptula et al., 2019), advocating for system-of-systems level engagements as the primary form of warfare in the information age and in future intelligent eras. Consequently, system-of-systems operations have emerged as a key aspect of military transformation worldwide. This approach emphasizes the effective utilization of networked connections among diverse entities to establish a cohesive and efficient combat system-of-systems (CSoS) (Gao et al., 2024, Koehler et al., 2024, Watson et al., 2023, Xu et al., 2024, Zhang et al., 2024), thereby capitalizing on informational advantages on the battlefield to achieve operational superiority. A CSoS (Li et al., 2024, Liu,et al., 2024, 2023, Markina-Khusid et al., 2021, Sun et al., 2024a, Uday and Marais., 2015) is a combat network composed of independently operating entities that interact with one another to provide overall operational capabilities, which cannot be achieved by a individual entities alone. In this network (Xu et al., 2023a), nodes represent entities, and links represent various interactions between entities.

The primary objective of studying CSoS is to ensure task fulfillment during operations (Sun et al.,2024b). As a type of long-lasting system, CSoS inevitably faces disturbances like enemy attacks, interference, or internal failures in dynamic combat environments, which can lead to performance degradation and failure to meet task requirements (Holling, 1973, Xu et al., 2024). Resilience (Artime et al., 2024, Du et al., 2023, Liu et al., 2022,

Wang et al., 2024a), defined as the system's ability to absorb and recover from disruptive events, offers a novel perspective for maintaining task fulfillment during operations. The profound uncertainties, complexities, and dynamics inherent in the battlefield present diverse challenges for the CSoS, underscoring the necessity to develop a more resilient CSoS. Moreover, the widespread implementation of advanced information and intelligent technologies further magnifies these challenges. The resilience of CSoS, defined as its ability to ensure task fulfillment during operations through the implementation of a series of recovery strategies, is closely related to its reliability (Chen et al., 2023b, Miller et al., 2021), effectiveness (Gao et al., 2023, Huang et al., 2023), and affordability (Chatterjee et al., 2023, 2022). Enhancing the resilience of CSoS is garnering increasing attention due to its practical value in optimizing network architectures, improving network security and refining operational planning.

The resilience analysis of CSoS is closely linked to network because of the inherent similarity between system-of-systems architecture and networks (De Domenico, 2023). Therefore, the first step in enhancing the resilience of CSoS is to establish an effective combat network model. Numerous researchers (Markina-Khusid et al., 2021, Chatterjee et al., 2022, Chen et al., 2024a, Filippini et al., 2014, Watson et al., 2022) contend that employing complex networks to characterize CSoS offers the most straightforward approach. However, the combat network integrates diverse entities that interact based on various functions, requiring a descriptive model beyond traditional complex networks to capture its heterogeneous nature. To address this issue, Li et al. (2019) introduced heterogeneous combat networks model to characterize CSoS, which takes into account various types of entities and information flows. Sun et al. (2023) employed a functional dependency network analysis method to capture interdependencies among heterogeneous entities and construct a dependent combat network. In addition to its heterogeneity, the CSoS exhibits distinct temporal characteristics, with its performance or structure evolving over time (Han et al., 2023, Uday et al., 2019, Xu et al., 2023b). To simultaneously capture the heterogeneity and dynamics of CSoS, Li et al. (2021) proposed a temporal combat network model that offered a precise description of its operational processes. Fang et al. (2021) analyzed the structure and operational processes of CSoS to construct a graphical evaluation and review technique (GERT) combat network, where entities serve as nodes and various interactions as directed links. System-of-systems and networks frequently exhibit spatial features (Lanier and Petnga, 2019, Rescia et al., 2023, Wang et al., 2024b, Yabe et al., 2022), and CSoS is no exception. However, existing combat network modeling and resilience optimization of CSoS have not sufficiently addressed this aspect. On one hand, the arrangement of entities within CSoS exhibits spatial features. For instance, AWACS must carefully balance proximity, which increases vulnerability to attack, and distance, which diminishes the effectiveness of intelligence gathering. On the other hand, interactions among entities also demonstrate spatial features. For example, UAVs are limited by remote control range; exceeding this threshold can lead to signal loss and reduced operational capability.

Similar to optimizing network resilience (Artime et al., 2024, Danziger. and Barabási, 2022, Edwards et al., 2024, Liu et al., 2022, Qi and Mei, 2024), current research on enhancing the resilience of CSoS primarily focuses on three aspects (Uday et al., 2019): pre-attack protection (Liu et al., 2024, Xing et al., 2019), mid-attack reconfiguration (Chen et al., 2023a, Li et al., 2024, Sun et al., 2024b, 2022, Tran et al., 2016), and post-attack recovery (Xu et al., 2023a, Zhong et al., 2024). Pre-attack protection primarily involves safeguarding critical entities or interactions within CSoS before an attack occurs, thereby reducing the negative impact of attacks on CSoS. For example, Liu et al. (2024) proposed an optimization method based on reinforcement learning to protect critical entities and enhance the resilience of CSoS under continuous attacks. Mid-attack reconfiguration involves reconfiguring interactions within CSoS without introducing external resources, aimed at mitigating the negative impact of attacks. For instance, Sun et al. (2022) presented a cooperative factor-based resilience optimization model that uses existing entities from one swarm to replace damaged entities in another swarm. However, in practice, pre-attack protection and mid-attack reconfiguration involve stringent conditions. The unpredictable and inevitable nature of enemy attacks makes it highly challenging to devise completely effective protection strategies beforehand. Furthermore, achieving interface interoperability among all entities in CSoS reconfiguration is somewhat idealistic and impractical. The two-year Russia-Ukraine conflict in 2022 tested that implementing post-disturbance recovery measures in CSoS is a straightforward method, including repairing infrastructure such as the Crimean Bridge, restoring damaged tank equipment, and recruiting new forces to meet task requirements. Post-attack recovery involves determining a set of entities requiring restoration and their recovery sequence (Aslani et al., 2024, Dui et al., 2024), with the aim of improving the performance of CSoS and thereby enhancing its resilience. Pan et al. (2019) and Xu et al. (2023a) respectively introduced recovery importance metrics from the perspectives of nodes and links within CSoS, aiming to recover critical components and thereby enhance the resilience of CSoS. Compared to these component importance measures, optimization methods offer a more efficient and effective approach to enhancing the resilience of CSoS. For example, Zhong et al. (2024) proposed a kill chain optimization method for resilient CSoS, focusing on optimizing the selection of kill chains within the

network to enhance resilience from a recovery perspective. However, existing research on post-attack recovery for CSoS emphasizes temporal aspects, overlooking its spatial features. During the recovery process in actual military operations, the physical distances between entities influence the sequence of entity restoration, as recovery teams take more time to move between distant combat entities. Few prior studies have incorporated the spatial features into enhancing post-attack recovery.

In view of these issues, this study explores the resilience enhancement problem for CSoS. By leveraging multilayer networks, operation loops, and network recovery optimization techniques, we present a unified framework called CSoS space-time resilience (CSoS-STRE) to integrate spatial features into the resilience analysis of CSoS. The CSoS-STRE framework comprises two key aspects: 1) the resilience optimization model and 2) the recovery optimization method. We summarize the main contributions of this study as follows.

i) We develop a spatial combat network model and a space-time resilience optimization model, which not only captures the complex spatial relationships between entities, accurately reflecting the operational process, but also reformulates the resilience enhancement problem as a classic linear optimization model with spatial features. Moreover, we extend the model from a no-obstacles scenario to one with obstacles, thereby further emphasizing spatial features.

ii) A resilience-oriented recovery optimization method based on an improved non-dominated sorting genetic algorithm (R-INSGA) is proposed to determine the optimal recovery sequence for the damaged entities. This method not only considers spatial features but also provides the optimal travel path for multiple recovery teams.

iii) Numerous experiments are conducted using a CSoS case study to demonstrate the feasibility, effectiveness, and superiority of the CSoS-STRE in enhancing the resilience of CSoS. By varying relevant parameters and comparing against baseline methods, the results provide valuable insights for guiding recovery and developing more resilient CSoS.

The structure of this study is outlined as follows. Section 2 describes the spatial combat network model, while Section 3 presents the spatial resilience optimization model and extends it to incorporate obstacles. In Section 4, we propose a recovery optimization method called R-INSGA. In Section 5, numerous experiments are conducted to demonstrate the feasibility, effectiveness and superiority of the proposed framework. Conclusions and future directions are presented in Section 6.

2 Spatial combat network model

2.1 Concept of operation loop

In system-of-systems operations, CSoS is required to effectively integrate various types of operational forces to meet task requirements. Typically, the CSoS comprises various types of entities with diverse functions such as sensing, control, interference, and strike capabilities. Based on Jeffrey's (Khasawneh et al., 2017), Dekker's (Deller et al., 2009) and Tan's (Li et al., 2017) combat model, entities in CSoS can be classified into four categories: sensor entities (S), decider entities (D), influencer entities (I), and enemy target entities (T). According to Boyd's OODA cycle theory (Yang et al., 2022), a basic combat process involves the sensor system detecting enemy targets (T→S) and transmitting relevant information to the decider entity (S→D). Subsequently, the decider entity issues commands to the influencer entity (D→I), which then exerts adverse effects on the enemy target entity (I→T).

The concept of operation loop can be employed to describe the aforementioned basic combat process. A basic operation loop consists of a sensor, a decider, an influencer, and an enemy target. Beyond this, there are three generalized types of operation loops. These multiple operation loops interconnect and intertwine with each other to form a CSoS. Figure 1 shows the four types of operation loops, which encompass all existing situations under current combat rules (Chen et al., 2023b, Li et al., 2024, Liu et al., 2024, Sun et al., 2024b).

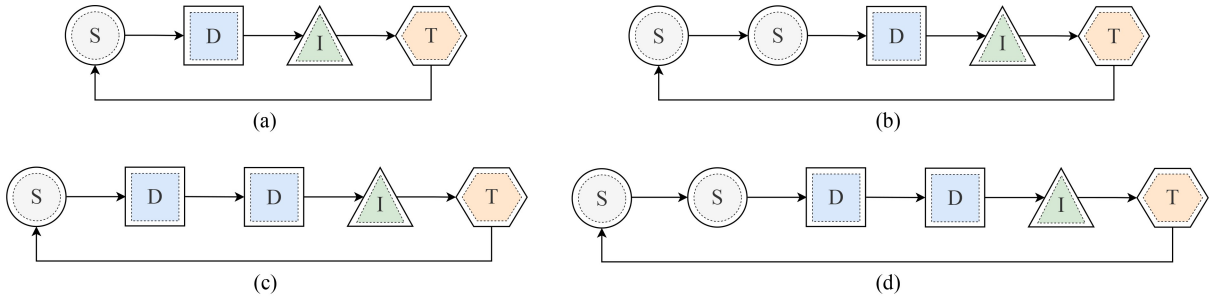

Fig. 1 Four different types of operation loops. (a) Basic operation loop: T→S→D→I→T. (b) Generalized operation loop: T→S→S→D→I→T. (c) Generalized operation loop: T→S →D→D→I→T. (d) Generalized operation loop: T→S→S→D→D→I→T.

2.2 Spatial combat network model of CsoS

Given that CSoS is influenced by spatial features during task completion (Chen et al., 2024b), and drawing inspiration from the concept of multilayer networks (De Domenico, 2023), we develop a spatial combat network model to characterize the spatial nature of CSoS. Non-interconnected networks, as multilayer networks without inter-layer links, effectively characterize multiple relationships within CSoS. A CSoS can be abstracted as a non-interconnected network $\mathcal{M}=(\mathcal{G},C)$, where $\mathcal{G}=\{G^{\alpha\in\{I,S\}}\}$ denotes the set of network layers, and C represents the set of inter-layer links. Here, $G^{[I]}$ depicts the information layer illustrating information flows between entities, while $G^{[S]}$ describes the spatial layer illustrating the distance relationships among these entities. Figure 2 illustrates an example of a spatial combat network. It is important to note that the entities in both layers are identical, and no interdependent links exist between the two layers.

In the information layer $G^{[I]}=(V^{[I]},E^{[I]})$, where node set $V^{[I]}=S\cup D\cup I\cup T=\{v_1^{[I]},v_2^{[I]},v_3^{[I]},\ldots,v_n^{[I]}\}$ represent the entities and link set $E^{[I]}=\{T\to S\}\cup\{S\to S\}\cup\{S\to D\}\cup\{D\to D\}\cup\{D\to I\}\cup\{I\to T\}=\{e_1^{[I]},e_2^{[I]},e_3^{[I]},\ldots,e_m^{[I]}\}$ denotes information flows between entities. Here, S,D,I,T represent sensor, decider, influencer, and target entities, respectively. The information flow $\{T\to S\}$, $\{S\to S\}$, $\{S\to D\}$, $\{D\to D\}$, $\{D\to I\}$, $\{I\to T\}$ denote target reconnaissance, reconnaissance sharing, information collection, decision-sharing, command issuance, and engagement with the enemy, respectively. If the number of nodes in the information layer is $n^{[I]}$, then the adjacency matrix $A^{[I]}$ is a square matrix of dimension $n^{[I]}\times n^{[I]}$, where $a_{ij}^{[I]}$ is represented as follows

$$a_{ij}^{[I]}=\begin{cases}1 & \left(v_i^{[I]},v_j^{[I]}\right)\in E^{[I]}\\ 0 & \text{otherwise}\end{cases},\tag{1}$$

where $i,j\in V^{[I]}$, and $i\neq j$.

In the spatial layer $G^{[S]}=(V^{[S]},E^{[S]})$, where node set $V^{[S]}=V^{[I]}$ and regarding the link set $E^{[S]}$, every pair of nodes is connected. In fact, the spatial layer is represented by a fully connected undirected graph. Differing from information layers, which are characterized by binary values 0 and 1 indicating entity connectivity, the spatial layer represents connections between entities using a distance measure d: larger distances between entities correspond to higher values of d. Assume the adjacency matrix $A^{[S]}$ is a square matrix of dimension $n^{[S]}\times n^{[S]}$, where $a_{ij}^{[S]}$ is represented as follows

$$a_{ij}^{[S]}=d_{ij}^{[S]},\left(v_i^{[S]},v_j^{[S]}\right)\in E^{[S]},\tag{2}$$

where $d_{ij}^{[S]}$ represents the distance between entities i and j.

When an entity in the network is damaged, it is removed from the information layer, leading to the removal of connected links. Conversely, if the entity is

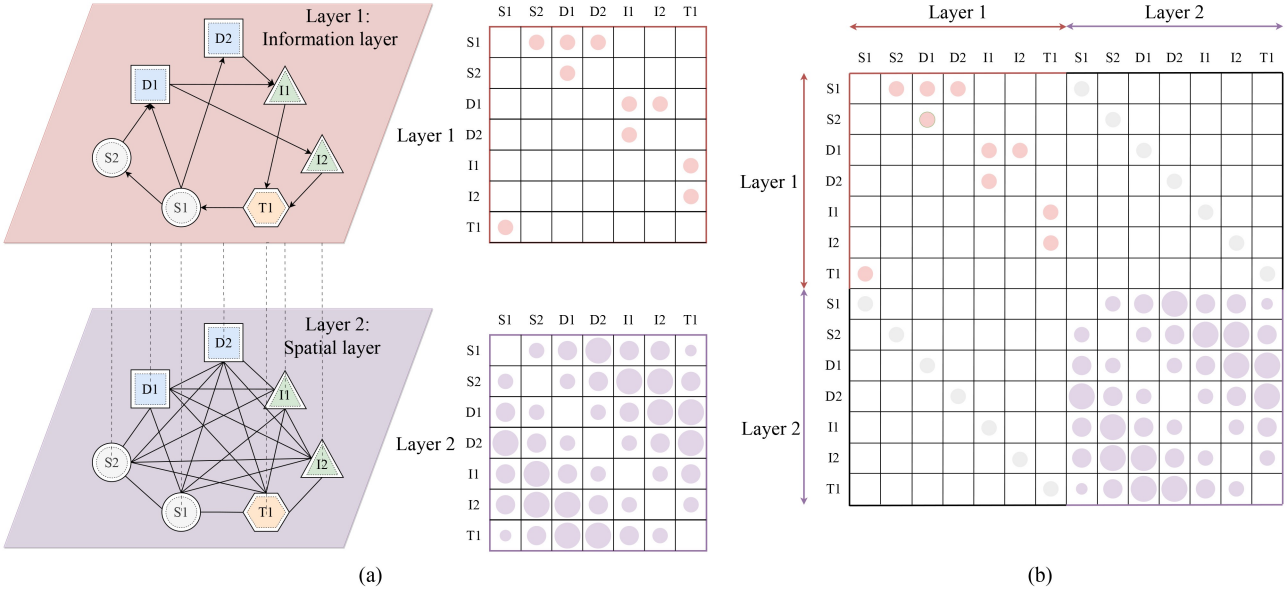

Fig. 2 The spatial combat network model. (a) The spatial combat network consists of the information layer (Layer 1) and the spatial layer (Layer 2). The former illustrates interaction relationships between entities, while the latter illustrates the distance relationships among these entities. In the information layer, the presence or absence of a pink circle indicates entity connectivity status. In the spatial layer, all entities are interconnected, and the area of the purple circle represents the distance between entities, with larger areas indicating greater distances. (b) The supra-adjacency matrix of the spatial combat network, where gray circulars represent projections from the information layer to the spatial layer.

restored, the opposite process occurs. Additionally, attacks and recoveries synchronously affect entities within the spatial layer. Let $V_{|\mathbb{K}|}$ denote the set of damaged entities and $E_{\circ}$ denote the corresponding damaged links. The spatial combat network, after the entity is destroyed, is represented by $\mathcal{G}=\{(V^{[I]}-V_{|\mathbb{K}|}, E^{[I]}-E_{\circ}),(V^{[S]}-V_{|\mathbb{K}|}, E^{[S]})\}$, where $|\mathbb{K}|$ denotes the number of entities to be damaged, with $|\mathbb{K}| \leqslant n$. Similarly, let $V^{|\mathbb{K}|}$ denote the set of restored entities and E° denote the corresponding restored links. The spatial combat network obtained after the entity restored is represented by $\mathcal{G}_{\gamma}=\{(V^{[I]}+V^{|\mathbb{K}|}, E^{[I]}+E^{\circ}),(V^{[S]}+V^{|\mathbb{K}|}, E^{[S]})\}$.

3 Spatial resilience optimization model for CSoS

3.1 The performance of CSoS

The performance of CSoS is achieved by aggregating the performance of all operation loops within its spatial combat network, which improves with the increasing number and capabilities of these operation loops. Typically, the capability of operation loops is assessed by quantifying both the length of the operation loop and the capability of entities involved. Referring to (Li et al., 2024, Li et al., 2021, Liu et al., 2024) the capability of the operation loop, $C(z_l)$, is calculated using Eq. (3):

$$C(z_l)=\frac{1}{|z_l|}\times C_T(v_t)\times\sum_{v_j\in V^S}C_S(v_j)\times \sum_{v_j\in V^D}C_D(v_j)\times\sum_{v_j\in V^I}C_I(v_j), \tag{3}$$

where z_l represents an operation loop involving a target entity v_t, a sensor entity set V^S, a decider entity set V^D and an influencer entity set V^I, the absolute value $|z_l|$ denotes the length of the operation loop z_l in the information layer. The symbols of C_S, C_D and C_I respectively represent the capability of sensor, decider, and influencer entities within the CSoS, while C_T denotes the importance of the target entity within the enemy's operational framework.

The capabilities of operation loop reflect the capability of attacking enemy target entities, while the number of operation loops signifies having more ways available for attacking these entities. Given the presence of multiple operation loops within a CSoS and varying importance levels of enemy target entities, the performance of CSoS, $P(\mathcal{M})$, is calculated using Eq. (4):

$$P(\mathcal{M})=\sum_{i\in V^T}\sum_{z_l\in Z_{\mathcal{M}}}w_i\times C(z_l), \tag{4}$$

where w_i denotes the weight of the i -th enemy target entity, with $i\in V^T$. The CSoS consists of a set of operation loop $Z_{\mathcal{M}}=\{z_l|l=1,2,\ldots,m\}$.

3.2 Resilience metrics

Resilience measurement is a crucial starting point in the study of CSoS. If a network collapse has already occurred, recovering the damaged entities proves to be an effective way to mitigate the declining performance of CSoS and enhance its resilience (Artime et al., 2024). Typically, the aim of this step is to establish the objective function to provide an optimized objective for the swift and efficient recovery of the CSoS during the recovery period. Therefore, our primary focus lies on two key aspects: recovery speed and performance. Drawing upon established resilience metrics in engineering systems (Cimellaro et al., 2010, Zobel, 2011), we utilize two specific metrics to measure the resilience of CSoS: recovery speed resilience $\mathcal{R}_s$ and cumulative performance loss resilience $\mathcal{R}_p$.

3.2.1 The recovery speed resilience $\mathcal{R}_s$

The recovery speed resilience $\mathcal{R}_s$ primarily quantifies the resilience of the CSoS from a temporal perspective, as shown in Eq. (5):

$$\mathcal{R}_s=\begin{cases}\dfrac{t_{\max}-t_{\chi}}{t_{\max}}, & t_{\chi}\leqslant t_{\max}\\ 0, & t_{\chi}>t_{\max}\end{cases}, \tag{5}$$

where $\mathcal{R}_s\in[0,1]$, a larger $\mathcal{R}_s$ signifies a faster recovery speed of the CSoS and indicates higher resilience. As shown in Fig. 3(b), t_{χ} and $t_{\max}$ respectively represent the time required to recover all damaged entities and the given maximum recovery time. Here, $t_{\max}$ is subjectively determined by decision-makers based on the urgency of the recovery task. $t_{\chi}=\sum_{i=1}^{k}t_x^i$, where t_x^i denotes the time required to recover the i -th damaged entity. When $t_{\chi}=0$, it indicates that CSoS instantly recovers after being attacked, with $\mathcal{R}_s=1$. When $t_{\chi}>t_{\max}$, it indicates that the total recovery time under the current recovery sequence (X) exceeds the given maximum recovery time deemed acceptable by the decision-makers, suggesting that the current recovery sequence is not feasible in terms of recovery speed, resulting in $\mathcal{R}_s=0$.

3.2.2 The cumulative performance loss resilience $\mathcal{R}_p$

In addition to temporal considerations, the resilience triangle model can be applied from a performance perspective to assess the resilience of CSoS, as illustrated in Eq. (6):

$$\mathbb{R}=\int_{T_r}^{T_r+t_{\chi}}(P(T_e)-P(T))\mathrm{d}T. \tag{6}$$

The value of $\mathbb{R}$ is inversely related to $\mathcal{R}_s$, with smaller

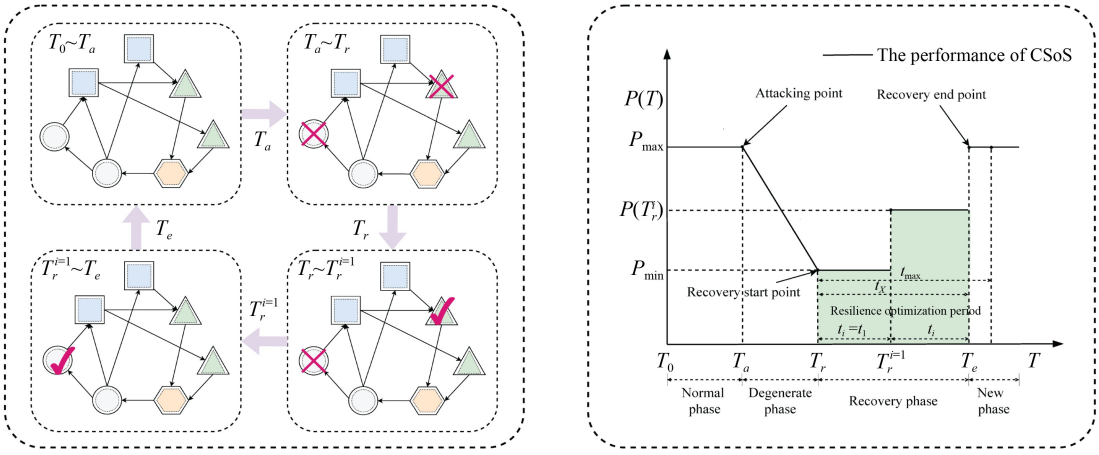

Fig. 3 The resilience enhancement process of CSoS. (a) Recovery activities involve symbols representing entities damaged by attacks, while true symbols denote the restoration of these entities. (b) The performance curve corresponding to recovery activities. The CSoS is attacked at time T_a and continues until time T_r, then begins recovery until time T_e ends. Our primary focus is on the T_r T_e period, the resilience optimization period (white area), to enhance the resilience of CSoS by identifying priority entities for restoration and determining their recovery sequence.

values being preferable. Consequently, we standardize $\mathbb{R}$ for analytical convenience to derive the cumulative performance loss resilience $\mathcal{R}_p$, as illustrated in Eq. (7):

$$\begin{aligned}\mathcal{R}_p &= 1-\frac{\int_{T_r}^{T_r+t_s}(P(T_s)-P(T))\mathrm{d}T}{t_{\max}\times P(T_s)}\\ &\approx 1-\frac{\sum_{i=1}^{k}(P(T_s)-P(T_r^i))\times t_x^i}{t_{\max}\times P(T_s)},\end{aligned} \tag{7}$$

where $T_r^0 < T_r^1 < T_r^2 < \cdots < T_r^k$, $T_r^0 = T_r$, $T_r^k = T_e = T_r + t_X$, $T_r^{i+1} - T_r^i = t_x^i$.

Graphically, $\mathcal{R}_p$ is quantified by the area of the white region, as shown in Fig. 3. A larger $\mathcal{R}_p$ indicates smaller cumulative performance loss, reflecting higher resilience. When $P(T_r^i) = P_{\min}$, it signifies that CSoS cannot recover. Consequently, without a resilience optimization period, resulting in $\mathcal{R}_p = 0$. When $P(T_r^i) = P(T_e)$, $\forall i \in k$, it indicates that the CSoS has experienced no post-attack loss, resulting in $\mathcal{R}_p = 1$.

3.3 Problem analysis

We use the example in Fig. 4 to illustrate the impact of different recovery strategies on CSoS by incorporating spatial features. In this example, each entity's capability is set to 1, and the performance of CSoS is calculated as $P(\mathcal{M}) = P(T_a) = 1.80/1.80 = 1$. Assuming that three entities (represented by false nodes) are damaged, the performance decreases from $P(\mathcal{M}) = P(T_a) = 1$ to $P(T_r = 0) = 0.25/1.80 = 0.14$. When spatial features are not considered (focusing solely on the information layer), there is no gap between the recovery of two damaged entities; the recovery of the next entity begins immediately after the previous one is completed. However, when spatial features are considered (with spatial layer present), a travel distance is required before the recovery of the next entity can begin after the previous entity has been restored. To further illustrate the impact of spatial features on the resilience of CSoS, we define two recovery strategies:

- Recovery strategy 1: The recovery sequence for the damaged entities is D1, I1, S2.
- Recovery strategy 2: The recovery sequence for the damaged entities is I1, D1, S2.

As shown in Fig. 4(b), when spatial features are not considered, both recovery strategies require the same amount of time to restore all entities. However, when spatial features are considered, recovery strategy 2 requires only 1.30 units of time to achieve maximum performance, whereas recovery strategy 1 requires 1.42 units of time. Since the required recovery time $t_X = \sum_{i=1}^{k} t_x^i$ is closely related to resilience metrics $\mathcal{R}_S$ and $\mathcal{R}_p$, it demonstrates that spatial features indeed affect the resilience of CSoS.

3.4 Resilience optimization model

Based on the description above, there is a significant difference between resilience optimization with spatial features and traditional resilience optimization in CSoS. Hence, this study categorizes the network recovery optimization problem for the resilience of CSoS as a network recovery optimization problem with spatial features and then presents a corresponding spatial resilience optimization model. Additionally, in real-world scenarios, specific areas may become impassable due to geographical factors or enemy blockades, referred to as 'obstacle'. Conse-

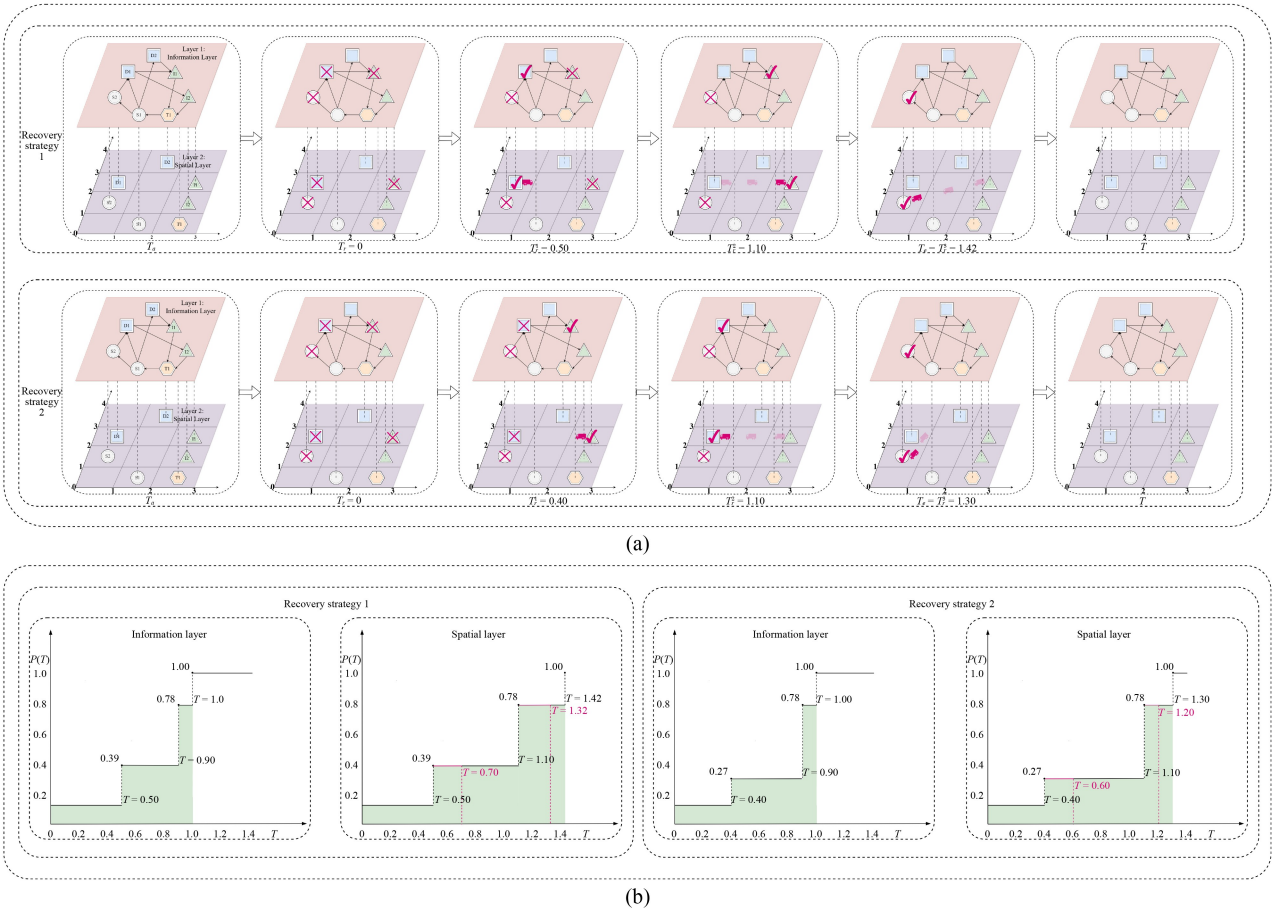

Fig. 4 The impact of different recovery strategies on the resilience of CSoS by incorporating spatial features. (a) A CSoS with seven entities, where each entity's capability is set to 1. The distance between two adjacent cells is 0.1 units, and the recovery team's (toy car) travel rate is 1 unit per time unit. The required recovery times for the damaged S1, D1 and I1 are set to 0.10, 0.50, and 0.40, respectively. Recovery strategies 1 and 2 correspond to recovery sequences D1→I1→S2 and I1→D1→S2, respectively. (b) The performance curves of the CSoS for the two recovery strategies. When spatial features are not considered, both recovery strategies require the same recovery time of 1 unit. When incorporating spatial features, the total travel distances for recovery strategy 1 and recovery strategy 2 are $0.2+\sqrt{(0.1^2+0.2^2)}=0.42$ and $0.2+0.1=0.30$, respectively. Consequently, the time required for recovery strategy 1 and recovery strategy 2 to achieve maximum performance is 1.42 units and 1.30 units, respectively.

quently, we extend the spatial resilience optimization to a new model that incorporates obstacles, termed the spatial resilience optimization model with obstacles. The notations in this section are listed in Table 1.

3.4.1 Spatial resilience optimization model

In an ideal scenario where recovery teams' travel path is unaffected by geographical factors or enemy blockades, meaning they can move freely. In this case, the shortest travel path between any two entities for the recovery teams is the straight-line distance between them. Based on this, we present the spatial resilience optimization model as follows.

$$\max\mathcal{R}_x=\begin{cases}\dfrac{t_{\mathrm{mx}}-t_\chi}{t_{\max}}, & t_\chi\leqslant t_{\max}\\ 0, & t_\chi>t_{\max}\end{cases}, \tag{8}$$

$$\max\mathcal{R}_p=1-\frac{\sum_{i=1}^{k}\left(P(T_e)-P(T_r^i)\right)\times t_x^i}{t(T_e)_{max}}, \tag{9}$$

s.t.

$$\mathbb{N}^i=\{n_{v_1}^i,n_{v_2}^i,n_{v_3}^i\ldots\},1\leqslant i\leqslant|\mathbb{N}|, \tag{10}$$

$$T_r^{i+1}-T_r^i=t_x^i,0\leqslant i\leqslant|\mathbb{K}|, \tag{11}$$

$$\bigcup_{i=1}^{i=|\mathbb{N}|}N^i=\mathbb{K};\ N^i\cap N^j=\varnothing,\text{ and }i\neq j, \tag{12}$$

$$\mathbb{T}_{v_1}^i=T_r^0+\frac{\min\{\mathbb{P}_i\to n_{v_1}^i\}}{\overline{V}_i}+t\left(n_{v_1}^i\right), \tag{13}$$

$$\mathbb{T}_{v_m}^i=\mathbb{T}_{v_{m-1}}^i+\frac{\min\{n_{v_{m-1}}^i\to n_{v_m}^i\}}{\overline{V}_i}+t\left(n_{v_m}^i\right),\quad 1\leqslant i\leqslant|\mathbb{N}|\text{ and }1<m\leqslant|\mathbb{K}|, \tag{14}$$

Table 1 Relevant parameter

Notation	Explanations
$t_{\max}$	The given maximum recovery time
t_X	The time required to restore all damaged entities
$\mathbb{N}$	The set of recovery team
N^i	The set of damaged entities restored by the i -th recovery team.
$n^i_{v_m}$	The i -th recovery team restores the m -th damaged entity.
$\mathbb{K}$	The set of damaged entities
$\mathbb{T}^i_{v_m}$	The moment when the i -th recovery team recovers the m -th damaged entity
$\mathbb{P}_i$	The initial position of the i -th recovery team
$\min\{\mathbb{P}_i \to n^i_{v_1}\}$	The minimum distance from the initial position of the i -th recovery team to the first damaged entity it recovers
$\overline{V_i}$	The average travel speed of the i -th recovery team
$t\left(n^i_{v_1}\right)$	The time taken by the i -th recovery team to recover the first damaged entity
$\min\{n^i_{v_{m-1}} \to n^i_{v_m}\}$	The minimum distance between the $(m-1)$ -th and m -th damaged entities, as recovered by the i -th recovery team
$\mathcal{T}^i$	The set of moments at which the i -th recovery team completes the recovery of damaged entities
$\mathbb{O}$	The set of obstacles
o_i	The i -th obstacle

$$\mathcal{T}^i = \{\mathbb{T}^i_{v_1}, \mathbb{T}^i_{v_2}\mathbb{T}^i_{v_3}, ...\}, \tag{15}$$

$$\{T^1_r, T^2_r, ...T^{|\mathbb{K}|}_r\} = \text{Sorted}\left\{\bigcup\nolimits_{i=1}^{i=|\mathbb{N}|}\mathcal{T}^i\right\}. \tag{16}$$

We consider various objective functions in our analysis. Specifically, the objective functions (8) and (9) represent the maximization of recovery speed resilience $\mathcal{R}_s$ and cumulative performance loss resilience $\mathcal{R}_p$, respectively. Equation (11) ensures that the number of recovered entities does not exceed the number of damaged entities, while Eq. (12) ensures that each damaged entity can be recovered only once by a single recovery team. Equation (13) represents the moment when the damaged entity v_1 is first restored by recovery team $\mathbb{N}^i$, while Eq. (14) indicate the moment when the damaged entity v_m is restored by recovery team $\mathbb{N}^i$. Equations (10) and (15) respectively represent the recovery sequence of damaged entities and the moment of recovery completion for each damaged entity in recovery team $\mathbb{N}^i$. Equation (16) aims to order the recovery completion times for each damaged entity. To further simplify, Eq. (16) can be reformulated as a combination of linear constraints, as follows:

$$T^i_r \leqslant T^{i+1}_r, T^i_r = \sum\nolimits_{j=1}^{j=|\mathbb{K}|} x_{i,j}\mathbb{T}_j,\ 1 \leqslant i \leqslant |\mathbb{K}|, \tag{17}$$

$$\sum\nolimits_{j=1}^{j=|\mathbb{K}|} x_{i,j} = 1,\ 1 \leqslant i \leqslant |\mathbb{K}|; \sum\nolimits_{i=1}^{i=|\mathbb{K}|} x_{i,j} = 1,\ 1 \leqslant j \leqslant |\mathbb{K}|, \tag{18}$$

$$x_{i,j} = 0 \text{ or } 1, 1 \leqslant i \leqslant |\mathbb{K}|,\ 1 \leqslant j \leqslant |\mathbb{K}|, \tag{19}$$

where variable $x_{i,j} = 1$ indicates that the damaged entity with identifier j is ranked i in the overall recovery sequence; $\mathbb{T}_j$ represents the moment at which the damaged entity with identifier j is restored. By introducing the variable $x_{i,j}$ along with Eqs. (17), (18) and (19), Eq. (16) essentially transforms the ordering problem into a 0-1 assignment problem.

3.4.2 Spatial resilience optimization model with obstacles

In real-world scenario, specific areas may become impassable due to geographical factors or enemy blockades. In this case, the travel paths of recovery teams are subject to these obstacles. The optimal path is no longer a straight line and must account for detours around these obstacles, as illustrated in Fig. 5. Clearly, the travel paths for recovery teams differ with and without obstacles, and Eqs. (13) and (14) are no longer applicable in the spatial resistance optimization model with obstacles. Therefore, the spatial resilience optimization model can be refined by accounting for obstacles, leading to an enhanced spatial resilience optimization model with obstacles, as detailed below.

$$\max\ \mathcal{R}_s, \tag{20}$$

$$\max\ \mathcal{R}_p, \tag{21}$$

s.t. Eqs. (10)–(19)

$$\begin{aligned}&\min\{n^i_{k,v_i} \to n^i_{k,v_{i+1}}\}\\ &= \min\{n^i_{v_{m-1}} \to n^i_{v_m} | \forall o_i \in \mathbb{O}, o_i \cap \{n^i_{v_{m-1}} \to n^i_{v_m}\} = \varnothing\},\end{aligned} \tag{22}$$

$$\min\{\mathbb{P}_i \to n^i_{v_1}\} = \min\{\mathbb{P}_i \to n^i_{v_1} | \forall o_i \in \mathbb{O}, o_i \cap \{\mathbb{P}_i \to n^i_{v_1}\} = \varnothing\}. \tag{23}$$

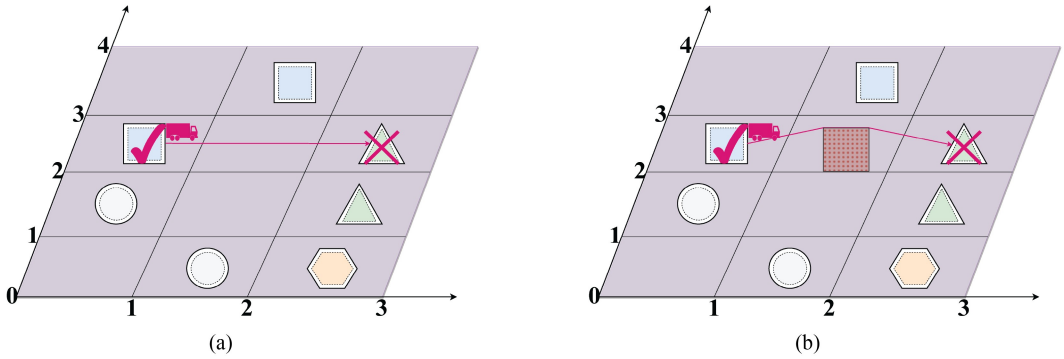

Fig. 5 Schematic diagram of optimal travel paths for recovery teams. (a) The optimal travel path for recovery teams in the absence of obstacles. Essentially, the optimal path is a straight-line distance between the damaged entities. (b) The optimal travel path for recovery teams in the presence of obstacles. In this case, the optimal path needs to navigate around and closely follow the obstacles, rather than being a straight-line distance.

The essential distinction between the two spatial resilience optimization models lies in the travel paths of the recovery teams, which influence the minimum distances between damaged entities. In the presence of obstacles, as outlined by Eqs. (22) and (23), both the travel paths of the recovery teams and the minimum distances between damaged entities are constrained by these obstacles, introducing additional spatial limitations.

4 The resilience-oriented recovery optimization method based on improved NSGA-II

4.1 Problem description and assumptions

The resilience optimization problem in CSoS presents a challenge due to the intricate nature of the complex spatial relationships between entities. The complexity is further compounded when obstacles in spatial combat networks and multi-objective optimization are considered, making the problem intractable for traditional methods. To overcome this, we propose a recovery optimization method called R-INSGA, which determines the optimal recovery sequence for damaged entities, thereby enhancing the resilience of CSoS.

Given the specific features of the resilience optimization problem in CSoS the following assumptions are made:

i) Upon damage, an entity's capability immediately drops to zero without intermediate states. Similarly, when the entity is restored, it follows the same principle.

ii) The model exclusively considers the scenario where entities are damaged, excluding cases where connections between entities are independently damaged. A link is assumed to be damaged if the entities it connects are damaged, and it recovers concurrently with the restoration of those entities.

iii) Complete information on all entities is known, including their type, capability, and location, as well as the location and size of any obstacles.

4.2 Solution algorithm

The NSGA-II (Deb et al., 2002, Hafsa et al., 2025), which introduces fast non-dominated sorting, has become a standard paradigm for multi-objective optimization problems and can effectively enhance the resilience of CSoS from both performance and time perspectives. Accordingly, we propose the R-INSGA method to enhance the resilience of CSoS. In addition, we introduce a variable neighborhood search (VNS) mechanism (Lozano-Osorio et al., 2025) with excellence and tabu criteria to the proposed R-INSGA. Specifically as follows.

1) Encoding and decoding.

Solutions are encoded according to the damaged entities, with each damaged entity corresponding to a gene (g_1, g_2), where g_1 represents the recovery team assigned to the damaged entity and g_2 indicates the recovery sequence of the damaged entity within the recovery team. For example, when $k_1:(g_1, g_2) = (2, 1)$, it indicates that the damaged entity k_1 is recovered by recovery team $\mathbb{N}^2$, and it ranks first in the recovery order of $\mathbb{N}^2$.

2) Mutation operator and selection operator.

We utilize a two-point mutation operator combined with a binary tournament selection operator.

3) Crossover operator.

The crossover operator randomly divides genes into two segments, designated as Segments A and B, as illustrated in Fig. 6(b). The gene of Child 1 is derived from Segment A of Parent 1 and Segment B of Parent 2, while the gene of Child 2 is derived from Segment B of Parent 1 and Segment A of Parent 2. Additionally, the crossover operator may produce infeasible solutions, which necessitate gene repair.

4) Variable neighborhood Search.

We designed two neighborhoods for the VNS, termed the 2-exchange neighborhood and the 3-exchange neighborhood, respectively. Each 2- or 3-exchange local search

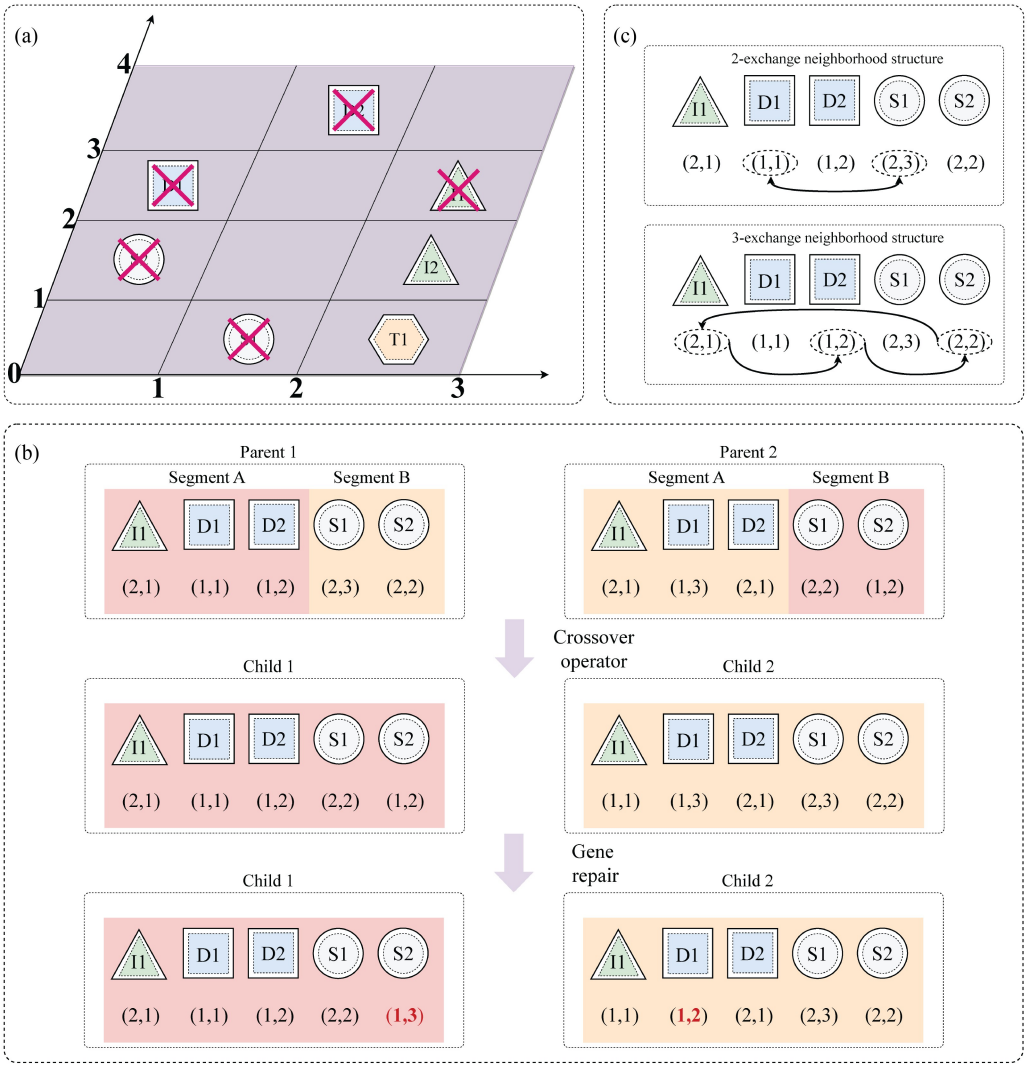

Fig. 6 The details of the R-INSGA for resilience enhancement in CSoS. (a) The CSoS $\mathcal{M}$ with five damaged entities. (b) Crossover operator with gene repair. The crossover operator randomly divides genes into two segments A and B. Segments A and B from Parent 1 and Parent 2 are exchanged to produce two offspring, Child 1 and Child 2. The genes obtained by Child 1 and Child 2 are highlighted with red and yellow backgrounds, respectively. Genes that change after repair are indicated in bold red, reflecting the modifications made to achieve feasible solutions. (c) Two neighborhood structures in VNS. In the 2-exchange neighborhood structure, two genes can be exchanged, while in the 3-exchange neighborhood structure, three genes can be exchanged.

randomly swaps the recovery teams and their repair sequences for two or three damaged entities, as illustrated in Fig. 6(c). To improve the effectiveness of the VNS, we introduce two additional criteria. The first is the excellence criterion, which applies VNS only to individuals in the Pareto front of each generation. The second is the tabu criterion, which excludes individuals that do not yield improved solutions from VNS in subsequent iterations, regardless of their Pareto front status.

Given the presence of obstacles, repair teams must navigate around these barriers to reach the damaged entities. The continuous Dijkstra algorithm (Wang, 2023) effectively addresses this challenge. Leveraging these approaches, the R-INSGA algorithm is formalized and presented in Algorithm 1.

5 Experimental analysis

5.1 Case description

To demonstrate the feasibility, effectiveness, and superiority of the CSoS-STRE in enhancing the resilience of CSoS, numerous experiments are conducted using a CSoS case study (Liu et al., 2024). The CSoS consists of 15 sensor entities, 10 decider entities, 20 influencer entities, and 5 enemy target entities. The topology of the CSoS is depicted in Fig. 7 and includes both the information layer and the spatial layer with and without obstacles.

5.2 Parameter settings

The detailed information of all entities in the space

Algorithm 1: Resilience-oriented recovery optimization method based on improved NSGA-II (R-INSGA)

Input: the details of CSoS $\mathcal{M}$, recovery teams $\mathbb{N}$, damaged entities $\mathbb{K}$ and obstacles $\mathbb{O}$;
Output: optimal travel paths, optimal recovery sequences, $\mathcal{R}_s$ and $\mathcal{R}_p$;
Initialize the R-INSGA parameters: population size, iteration number, mutation probability;
Randomly initialise the population $\mathcal{P}$;
foreach $p_i \in \mathcal{P}$ **do**
  if $\mathbb{O} \neq \varnothing$ **then**
    Solve the optimal travel path of p_i by continuous Dijkstra algorithm;
  end
  Calculate the $\mathcal{R}_s$ and $\mathcal{R}_p$ of p;
end
Sort $\mathcal{P}$ by fast non-dominated sort and crowding distance;
while (not stop criterion) **do**
  Use selection operator on $\mathcal{P}$ to select the next offspring' parents $\mathcal{P}_{\text{parent}}$;
  Use crossover operators on $\mathcal{P}_{\text{parent}}$ to generate offspring populations $\mathcal{P}_{\text{child}}$;
  Use mutation operators on offspring populations $\mathcal{P}_{\text{child}}$;
  foreach $p' \in \mathcal{P}_{\text{child}}$ **do**
    if $\mathbb{O} \neq \varnothing$ **then**
      Solve the optimal travel path of p_i' by continuous Dijkstra algorithm;
    end
    Calculate the $\mathcal{R}_s$ and $\mathcal{R}_p$ of p_i';
  end
  $\mathcal{P}_{\text{merge}} = \mathcal{P} \cup \mathcal{P}_{\text{child}}$;
  Sort $\mathcal{P}_{\text{merge}}$ by fast non-dominated sort crowding distance;
  Perform VNS on the population $\mathcal{P}_{\text{merge}}$ based on excellence and tabu criteria;
  Sort $\mathcal{P}_{\text{merge}}$ by fast non-dominated sort crowding distance;
  Execute elitism strategy for population $\mathcal{P}_{\text{merge}}$ to obtain the next generation $\mathcal{P}_{\text{next}}$;
  $\mathcal{P} = \mathcal{P}_{\text{next}}$;
end

combat network are shown in Table 2. The space combat network comprises 321 operation loops in the information layer, while the spatial layer is defined as an "4200 × 3500" area with five obstacles. The CSoS comprises 50 entities and 176 links, with an attack event resulting in the destruction of 15 entities. Currently, there is one recovery team ($|\mathbb{N}| = 1$), with its travel speed is set to $\overline{V} = 50$. The maximum recovery time is set to $t_{\max} = 1500$.

5.3 Resilience analysis

Assuming that the CSoS experiences random attacks resulting in the destruction of 15 entities (S8, S13, D1, D2, D3, D5, D6, D7, I2, I6, I10, I12, I18, I19, I20), its performance reaches its lowest point at $T_r = 0$, and performance drops from $Pa_{\max}$ to $P(T_r) = 4.189$. To demonstrate the feasibility and effectiveness of the proposed framework, we compare it with traditional random recovery strategies (RRS). In this analysis, we consider a single recovery team and evaluate its performance from four different initial positions: down (D5), right (D1), up (I6), and left (I8).

Figure 8(a) depicts the travel path of the recovery team in the absence of obstacles. As shown in Fig. 8, regardless of the recovery team's initial position, the curve of the proposed R-INSGA consistently outperforms that of RRS. Similarly, in the presence of obstacles (Fig. 9), the curve for the proposed R-INSGA remains superior to that of RRS. However, with obstacles present, the travel path

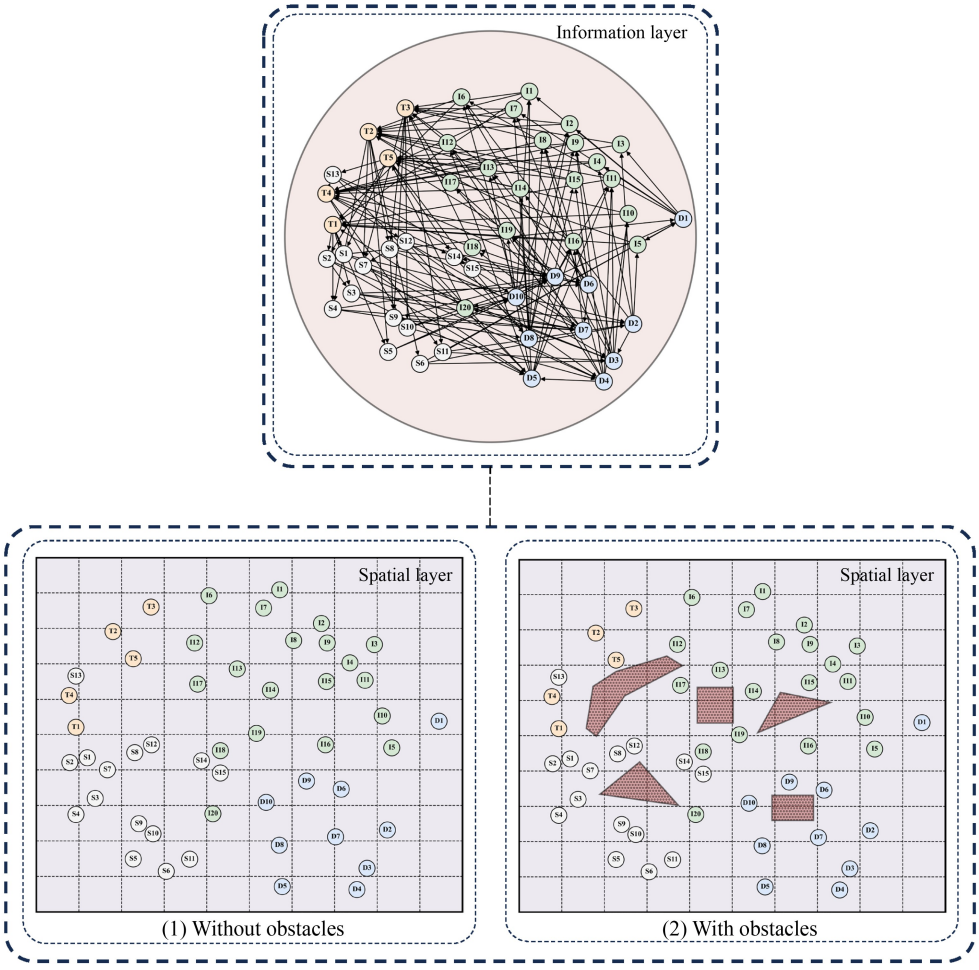

Fig. 7 The spatial combat network without and with obstacles.

Table 2 The detailed information regarding entities within the CSoS

No.	Entity	Capability	Recovery time	Weight	Location
1	T1	0.70	\	0.95	(3390, 1920)
2	T2	0.72	\	0.90	(3755, 2868)
3	T3	0.82	\	0.85	(4130, 3110)
4	T4	0.80	\	0.85	(3323, 2238)
5	T5	0.66	\	0.80	(3955, 2600)
6	S1	0.86	60	\	(3502, 1620)
7	S2	0.58	36	\	(3330, 1572)
8	S3	0.88	36	\	(3580, 1219)
...	...	...	...	\	...
43	I13	0.80	48	\	(4980, 2500)
44	I14	0.70	48	\	(5308, 2290)
45	I15	0.76	36	\	(5860, 2376)
46	I16	0.80	60	\	(5850, 1748)
47	I17	0.78	42	\	(4590, 2350)
48	I18	0.66	42	\	(4814, 1690)
49	I19	0.74	54	\	(5170, 1860)
50	I20	0.86	48	\	(4740, 1068)

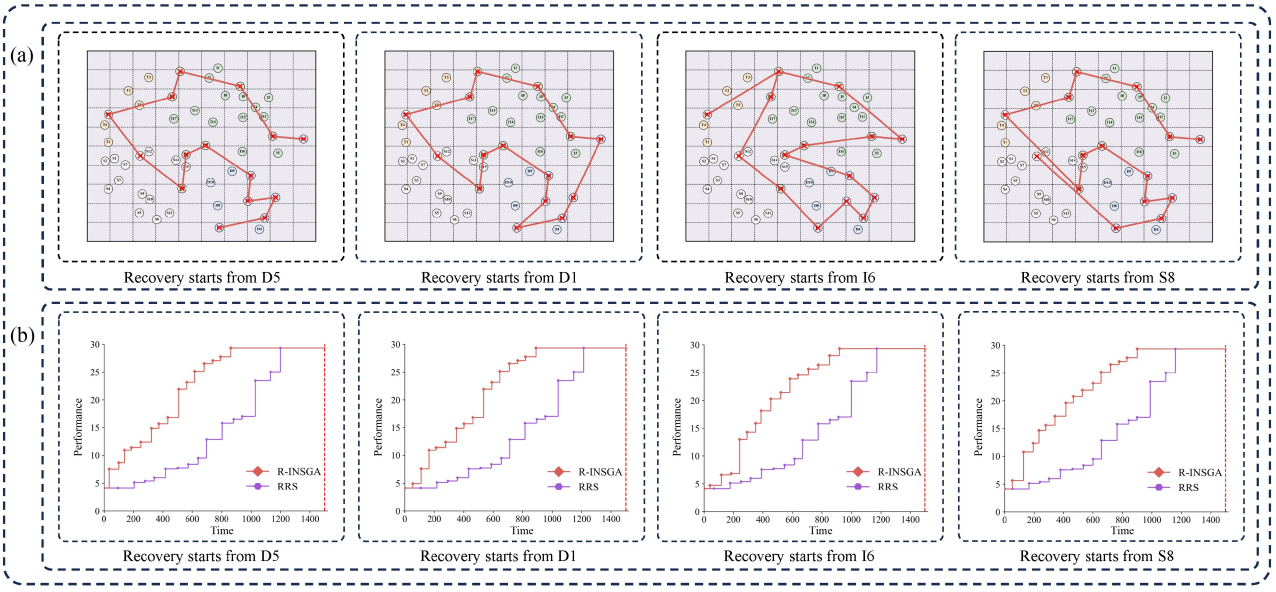

Fig. 8 The travel paths of the recovery team and the performance curves of the CSoS in the absence of obstacles.

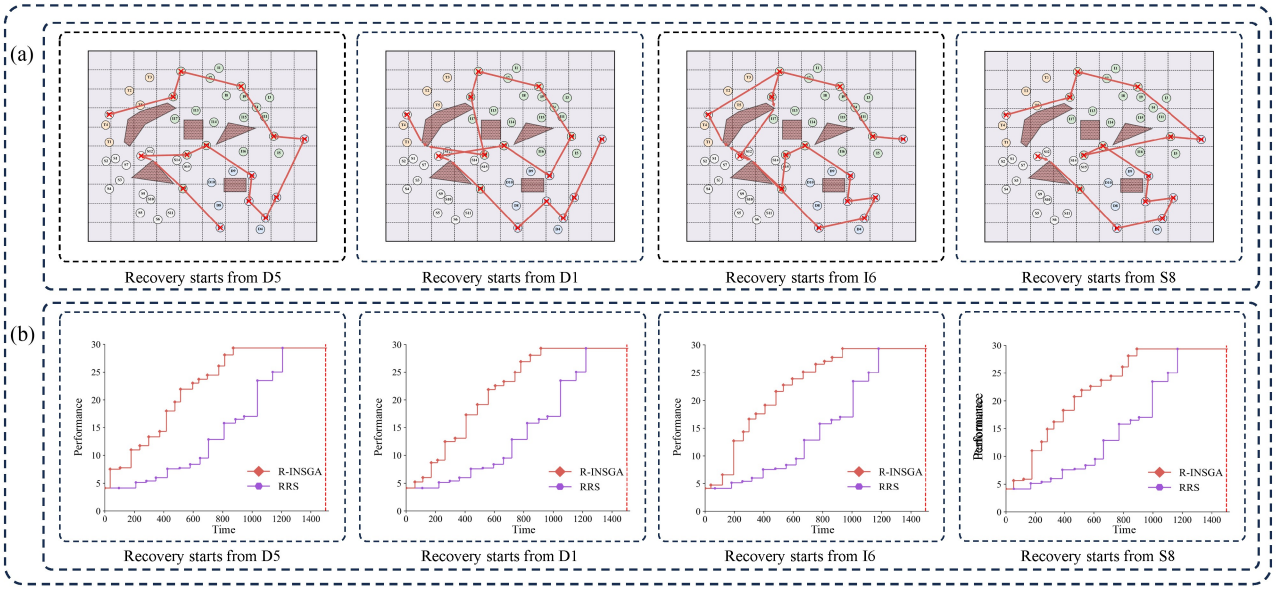

Fig. 9 The travel paths of the recovery team and the performance curves of CSoS in the presence of obstacles.

of the recovery team becomes more complex due to the inability to navigate directly through the obstacles.

To facilitate the comparison of resilience, we define the resilience comparison parameter as $\overline{\mathcal{R}} = (\mathcal{R}_s + \mathcal{R}_p)/2$. As shown in Table 3, the proposed R-INSGA demonstrates superior performance compared to RRS in enhancing the resilience of CSoS. For example, with initial positions at D5 and obstacles present, the proposed R-INSGA achieves a 69% greater resilience enhancement compared to RRS. In addition, we observe that, overall, although the resilience values with obstacles are not significantly different from those without obstacles, it remains evident that the resilience enhancement effect is weaker in the presence of obstacles. The former is due to the given maximum recovery time $t_{\max}$ being excessively long, while the latter is attributable to the longer travel path of the recovery team in the presence of obstacles. The above findings align with real-world situations, demonstrating that the proposed space combat network and resilience optimization model are both feasible and effective.

5.4 Extend analysis

5.4.1 Effect of the number of recovery team

Keeping other parameters constant, Figs. 10 and 11 illustrate the effect of the number of recovery teams on the travel path in the presence of obstacles. Here, we

Table 3 Comparison of resilience values between the proposed method and the RRS under different scenarios

Scenario		$\mathbb{P}(D5)$			$\mathbb{P}(D1)$			$\mathbb{P}(I6)$			$\mathbb{P}(I8)$		
		$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$	$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$	$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$	$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$
No-obstacles	R-INSGA	0.42	0.77	0.60	0.41	0.75	0.58	0.39	0.75	0.57	0.40	0.77	0.59
	RRS	0.20	0.51	0.35	0.19	0.50	0.35	0.22	0.52	0.37	0.23	0.53	0.38
Obstacles	R-INSGA	0.42	0.75	0.59	0.39	0.73	0.56	0.38	0.76	0.57	0.41	0.76	0.58
	RRS	0.20	0.50	0.35	0.19	0.50	0.34	0.21	0.52	0.37	0.22	0.53	0.37
Scenario(obstacles)		$\lvert\mathbb{N}\rvert = 1$			$\lvert\mathbb{N}\rvert = 2$			$\lvert\mathbb{N}\rvert = 3$			$\lvert\mathbb{N}\rvert = 4$		
		$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$	$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$	$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$	$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$
Same direction	R-INSGA	0.42	0.77	0.60	0.70	0.87	0.79	0.79	0.91	0.85	0.83	0.92	0.88
	RRS	0.20	0.51	0.35	0.60	0.78	0.69	0.68	0.86	0.77	0.74	0.89	0.82
Different directions	R-INSGA	0.42	0.75	0.59	0.69	0.88	0.79	0.80	0.91	0.86	0.85	0.93	0.89
	RRS	0.20	0.50	0.35	0.42	0.75	0.58	0.66	0.84	0.75	0.70	0.86	0.78
Scenario(obstacles)		$t_{\max}$			$t_{\max}$			$t_{\max}$			$t_{\max}$		
		$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$	$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$	$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$	$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$
Different directions	R-INSGA	0.85	0.93	0.89	0.78	0.91	0.84	0.62	0.83	0.73	0.00	0.31	0.16
	RRS	0.70	0.86	0.78	0.63	0.82	0.72	0.18	0.59	0.38	0.00	0.07	0.03
Scenario(obstacles)		$\overline{V} = 50$			$\overline{V} = 40$			$\overline{V} = 30$			$\overline{V} = 20$		
		$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$	$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$	$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$	$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$
Different directions	R-INSGA	0.62	0.83	0.73	0.61	0.81	0.71	0.55	0.82	0.68	0.48	0.80	0.64
	RRS	0.24	0.66	0.45	0.27	0.56	0.42	0.00	0.48	0.24	0.00	0.48	0.24

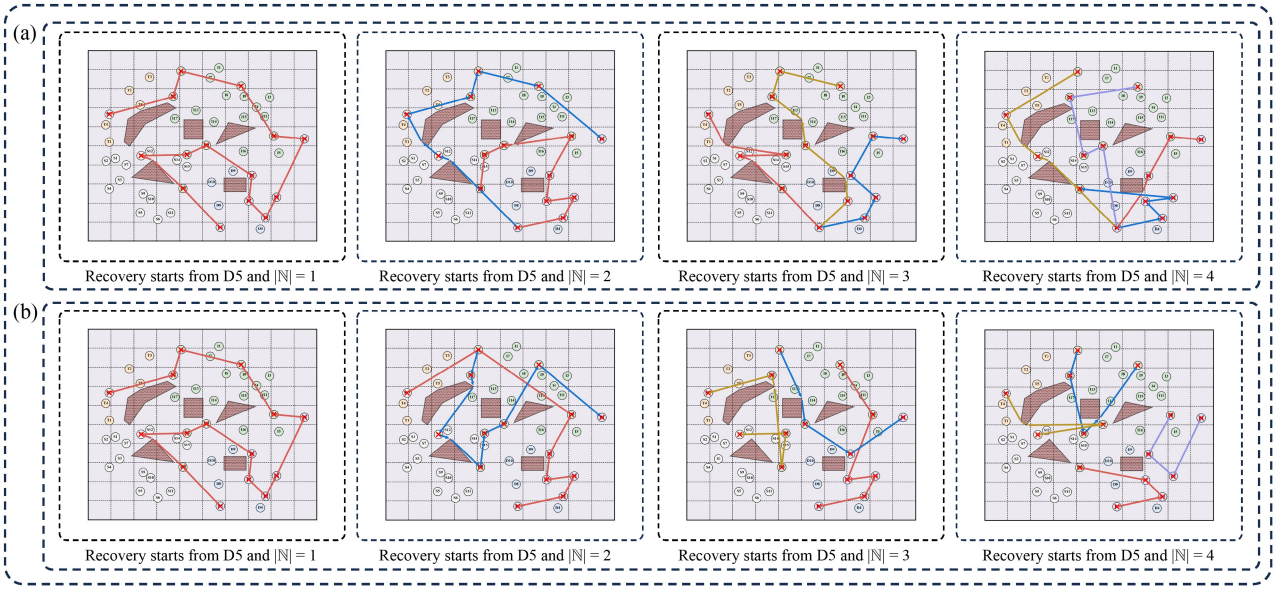

Fig. 10 The travel paths for different numbers of recovery teams.

considered two scenarios for four different numbers of recovery teams: one where all recovery teams start from position D5 (same direction), and another where the recovery teams start from different positions (different directions).

Table 3 provides the resilience enhancement effects, indicating that a single recovery team ($|\mathbb{N}| = 1$) is insufficient, and the resilience enhancement effect gradually improves. Note that variations in $|\mathbb{N}|$ and $\overline{\mathcal{R}}$ are not linearly related. For example, in the case of different directions, the resilience value $\overline{\mathcal{R}}$ for $|\mathbb{N}| = 2$ is 33.90% higher compared to $|\mathbb{N}| = 1$, while $|\mathbb{N}| = 3$ shows only an 8.86% improvement over $|\mathbb{N}| = 2$, and $|\mathbb{N}| = 4$ provides a 3.49% increase over $|\mathbb{N}| = 3$. In addition, the impact of

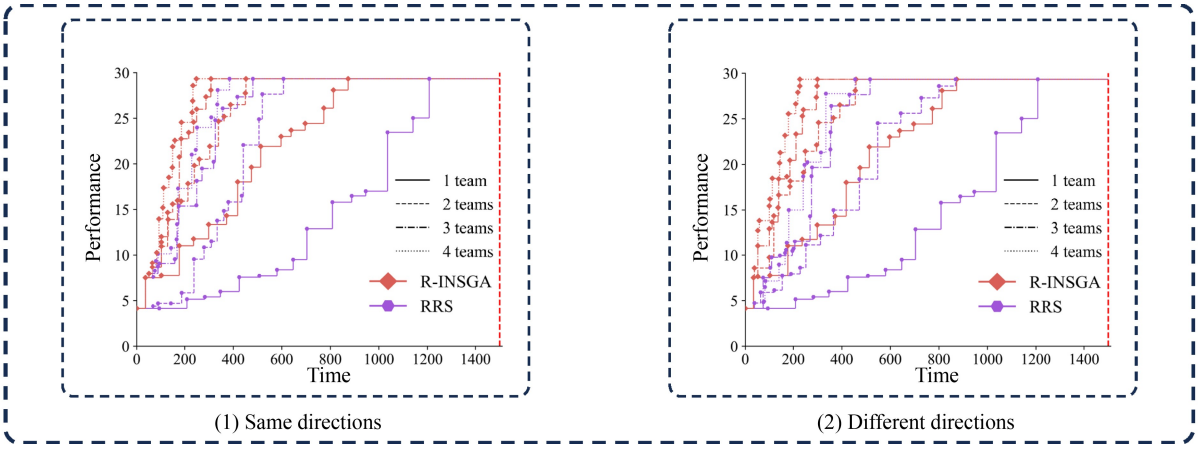

Fig. 11 The performance curves of the CSoS under different initial positions.

the recovery teams' initial positions on resilience values is minimal between the same direction and different directions scenarios. This is due to the fact that the given maximum recovery time and other related recovery resources have been set to sufficient levels.

The above experiment yields the following insights: indiscriminately increasing the number of recovery teams does not necessarily lead to significantly better outcomes. It is crucial to balance the costs of adding more teams with the benefits of improved resilience. Additionally, when recovery resources are set to sufficient levels, decision-makers do not need to consider the initial positions of multiple recovery teams.

5.4.2 Effect of the given maximum recovery time

Keeping other parameters constant, Figs. 12 and 13 illustrate the effect of given maximum recovery time on resilience, considering obstacles and different directions. We consider four scenarios with different maximum recovery times: $t_{\max} = 1500, 1050, 600, 150$.

As shown in Fig. 12 and Table 3, when the given maximum recovery time $t_{\max} \geqslant 226$, variations in $t_{\max}$ do not alter the travel paths or recovery sequence, as the time required to recover all damaged entities is $t_X = 226$. However, when $t_{\max} = 150 < 226$, meaning that the time to restore all damaged entities exceeds the given maximum recovery time, the travel path and recovery sequence of the recovery teams will change. This is because recovery teams cannot recover all damaged entities; rather, they can only restore a specific subset of damaged entities within a given maximum recovery time to achieve a locally optimal resilience value rather than a global optimum. According to Eqs. (5) and (7), $\mathcal{R}_s$ and $\mathcal{R}_p$ are positively correlated with $t_{\max}$, and thus increase as $t_{\max}$ increases. Therefore, $t_{\max}$ can be determined based on practical considerations to highlight the urgency of recovery.

The above experiment yields the following insights: decision-makers can adjust the travel paths of recovery teams to achieve local optimization by altering the given maximum recovery time. Additionally, setting the given maximum recovery time also reflects the urgency of the recovery effort.

5.4.3 Effect of the travel speed of recovery team

Keeping other parameters constant, Figs. 13 and 14 illustrate the effect of the travel speed of recovery team on resilience, considering obstacles and different directions. We consider four situations for the travel speed of recovery teams: $\overline{V} = 50, 40, 30, 20$.

As shown in Fig. 13 and Table 3, travel speed significantly affects both the recovery sequence of damaged entities and the effectiveness of resilience enhancement. As travel speed $\overline{V}$ increases, the time required to recover all damaged entities t_X decreases and the resilience value $\overline{\mathcal{R}}$ improves. For example, at a travel speed of $\overline{V} = 40$, $\overline{\mathcal{R}} = 0.73$, whereas at a travel speed of $\overline{V} = 20$, $\overline{\mathcal{R}} = 0.68$. The former is 7% more effective than the latter. Essentially, $\overline{V}$ represents the recovery speed rate to some extent. If the travel speed is too low, it may not be possible to restore all damaged entities within the given maximum recovery time $t_{\max}$. Conversely, if the travel speed is too high, the associated recovery costs may become prohibitive, as travel speed is often closely linked to recovery costs in practical scenarios. Therefore, $\overline{V}$ can be determined based on practical recovery cost considerations.

The above experiment yields the following insights: adjustments to travel speed directly influence the rate of recovery, with higher travel speeds accelerating the overall recovery process. Decision-makers can optimize the trade-off between recovery costs and resilience enhancement by adjusting $\overline{V}$ in accordance with practical constraints and operational considerations. For the extended analysis of the no-obstacle scenario, please refer to Supplementary Material 1. The results adhere to the same rules as those

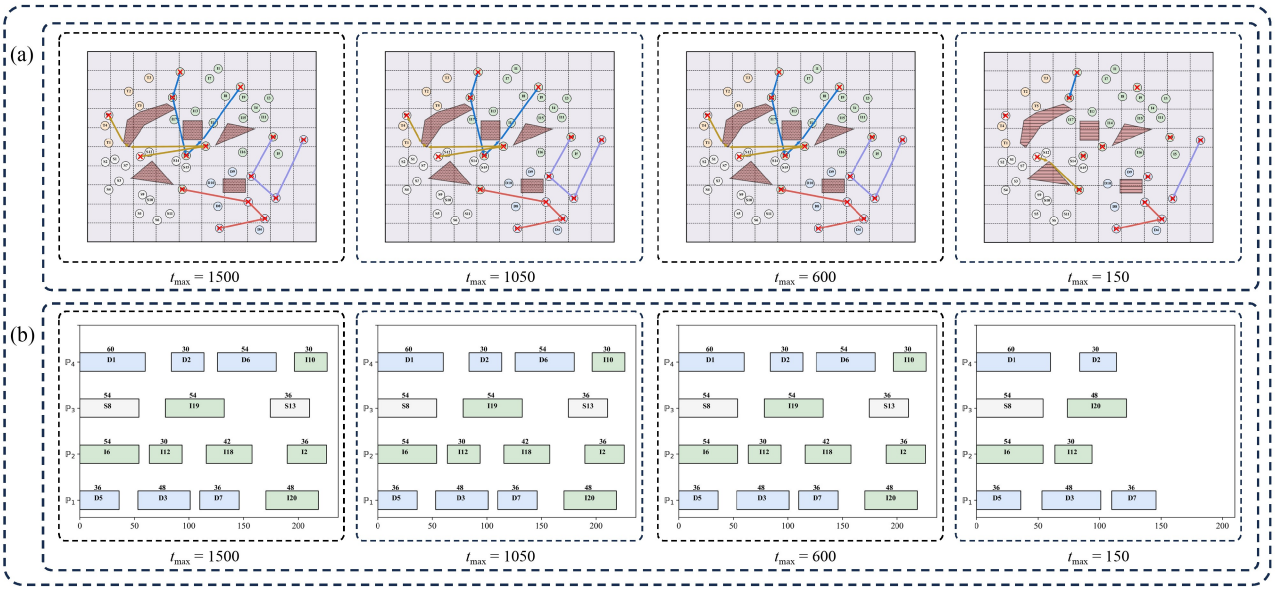

Fig. 12 The travel paths and recovery sequence for different given maximum recovery time.

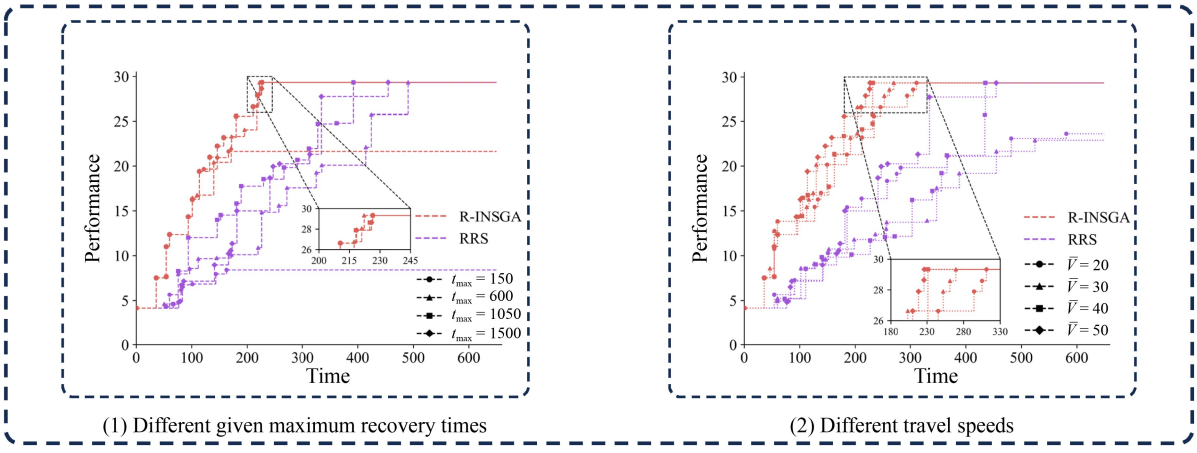

Fig. 13 The performance of the CSoS under different maximum recovery times and different travel speeds.

observed in the obstacle scenario's extended analysis.

5.5 Baseline comparison

To demonstrate the superiority of the proposed framework, we replace the previous RRS with two deliberate attack strategies, each addressing different perspectives. i) Deliberate attack 1: this strategy employs a degree centrality-based approach, where entities in the CSoS are sequentially removed in descending order of their degree centrality, focusing on the structural perspective. ii) Deliberate attack 2: this strategy uses a maximum operation loop-performance-first approach, removing entities in descending order based on their impact on the performance of operation loops, thereby concentrating on the functional perspective. Additionally, we select six baseline methods (Liu et al., 2024) and RRS for comparison with the proposed framework. The methods are described as follows.

1) Performance-based recovery strategy (PBRS). The damaged entities are restored in descending order based on their instantaneous performance improvement.

2) Time-based recovery strategy (TBRS). The damaged entities are restored in ascending order based on the speed of performance improvement. For example, if restoring damaged entity A takes less time compared to restoring damaged entity B, entity A will be prioritized for recovery.

3) Degree centrality-based recovery strategy (DCRS). The damaged entities are restored in descending order of their degree centrality values.

4) Betweenness centrality-based recovery strategy (BCRS). The damaged entities are restored in descending order of their betweenness centrality values.

5) Closeness centrality-based recovery strategy (CCRS). The damaged entities are restored in descending

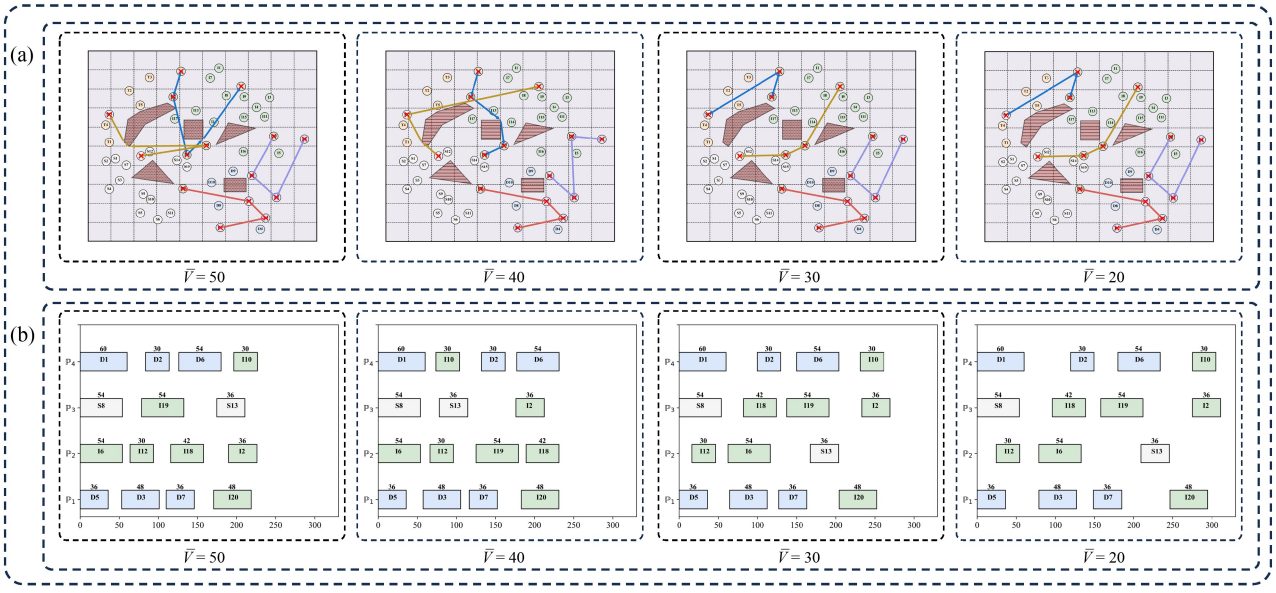

Fig. 14 The travel paths and recovery sequences for different travel speeds.

order based on their closeness centrality values.

6) Eigenvector centrality-based recovery strategy (ECRS). The damaged entities are restored in descending order of their eigenvector centrality values.

We consider both single recovery team ($|\mathbb{N}|=1$, $t_{\max}=1500$) and multiple recovery team ($|\mathbb{N}|=4$, $t_{\max}=600$) scenarios for a comprehensive analysis, with Fig. 15(a) illustrating the travel paths of under the proposed R-INSGA. As shown in Fig. 15(b) and Table 4, the performance curve of R-INSGA generally positions above those of the other baseline methods, regardless of the attack strategy or the number of recovery teams, indicating its superiority. For instance, under deliberate attack 1, a single recovery team restores all damaged entities more quickly than other recovery strategies (except TBRS), achieving full restoration in just 838.88 time units. Surpassing other recovery strategies in resilience values by 39.39%, 3.86%, 20.72%, 17.23%, 50.70%, and 52.45%, respectively. Similarly, in the scenario with four recovery teams, it is evident that R-INSGA outperforms other recovery strategies in both $\mathcal{R}_s$ and $\mathcal{R}_p$. Based on the values of $\overline{\mathcal{R}}$, the ranking of the recovery strategies is R-INSGA > TBRS > > PBRS > DCRS = BCRS > ECRS > CCRS > RRS, underscoring the superiority of the proposed framework.

6 Conclusion and future directions

Nowadays, modern warfare has transitioned from the paradigm of isolated combat forces to system-to-system confrontations. However, CSoS inevitably faces disturbances like enemy attacks, interference, or internal failures in dynamic environments, which can lead to performance degradation and failure to meet task requirements. Enhancing the resilience of CSoS is garnering increasing attention due to its practical value in optimizing network architectures, improving network security, and refining operational planning. To address these challenges, we present a unified framework called CSoS-STRE, which incorporates spatial features to enhance CSoS resilience, offering the following benefits.

i) CSoS-STRE fully integrates spatial features and represents CSoS as multilayer networks, including information and spatial layers, offering a precise depiction of the operational process.

ii) CSoS-STRE not only incorporates practical factors such as the given maximum recovery time, as well as the number and travel speed of recovery teams, but also skillfully reformulates the problem into a linear optimization model.

iii) CSoS-STRE can simultaneously provide optimal travel paths and recovery sequences. Compared to other baseline methods, it achieves faster convergence and better performance.

The results provide valuable insights for guiding recovery and developing more resilient CSoS. Some insights are as follows: i) indiscriminately increasing the number of recovery teams does not necessarily lead to significantly better outcomes. ii) decision-makers can adjust the travel paths of recovery teams to achieve local optimization by altering the given maximum recovery time. iii) adjustments to travel speed directly influence the rate of recovery, with higher travel speeds accelerating the overall recovery process. While this study represents a foundational step in enhancing the resilience of CSoS through the integration of spatial features, several challenges remain to be addressed in future research. For example, leveraging machine learning to develop more efficient and cost-effective optimization algorithms offers a promising

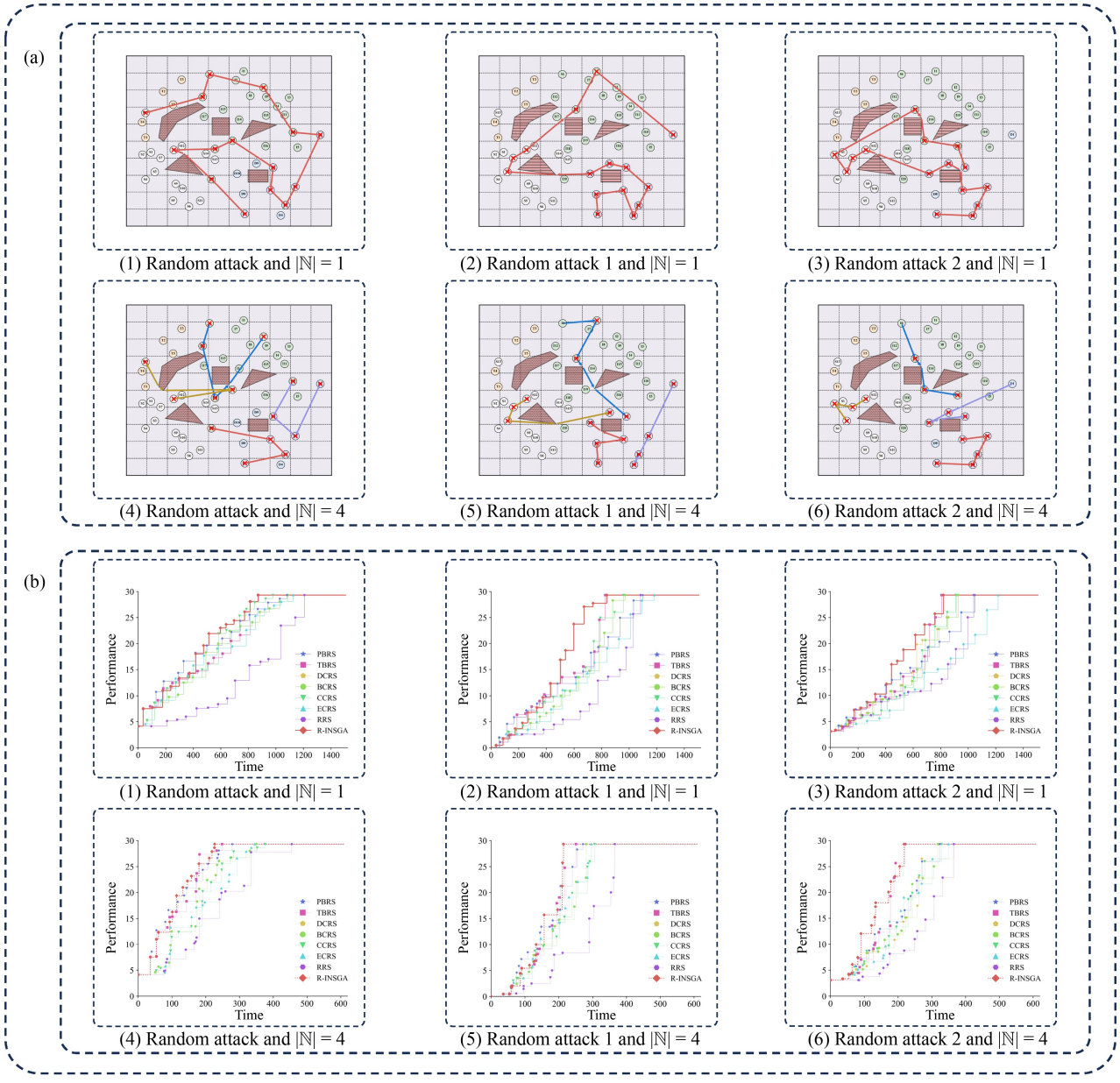

Fig. 15 Comparative analysis of the proposed method and baseline method under different strategies with obstacles.

Table 4 Resilience results of CSoS under different strategies with obstacles.

Attack strategy (Damaged entities)	Recovery strategy	$\|\mathbb{N}\| = 1$ ($\mathbb{P}$(D5))				$\|\mathbb{N}\| = 4$ (Different directions)			
		$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$	t_X	$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$	t_X
Random attack (S8, S13, D1, D2, D3, D5, D6, D7, I2, I6, I10, I12, I18, I19, I20)	PBRS	0.25	0.74	0.50	1124.72	0.43	0.82	0.62	344.82
	TBRS	0.42	0.72	0.57	869.58	0.59	0.82	0.70	248.59
	DCRS	0.27	0.70	0.49	1087.85	0.42	0.75	0.59	350.25
	BCRS	0.27	0.70	0.49	1087.85	0.42	0.75	0.59	350.25
	CCRS	0.35	0.72	0.54	977.27	0.37	0.74	0.55	375.21
	ECRS	0.25	0.69	0.47	1124.72	0.43	0.72	0.57	344.82
	RRS	0.20	0.50	0.35	1207.22	0.24	0.66	0.45	454.69
	R-INSGA	0.42	0.75	**0.59**	872.98	0.62	0.83	**0.73**	226.44

(*Continued*)

Attack strategy (Damaged entities)	Recovery strategy	$\lvert\mathbb{N}\rvert = 1$ ($\mathbb{P}$(D5))				$\lvert\mathbb{N}\rvert = 4$ (Different directions)			
		$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$	t_X	$\mathcal{R}_s$	$\mathcal{R}_p$	$\overline{\mathcal{R}}$	t_X
Deliberate attack 1 (D9, D3, D5, D1, D4, D7, D8, D2, D6, D10, I13, S3, S7, S8, I1)	PBRS	0.21	0.60	0.40	1181.91	0.51	0.72	0.61	295.68
	TBRS	0.45	0.64	0.54	826.01	0.58	0.71	0.64	250.21
	DCRS	0.35	0.58	0.47	941.48	0.53	0.67	0.60	281.00
	BCRS	0.35	0.58	0.47	941.48	0.53	0.67	0.60	281.00
	CCRS	0.36	0.60	0.48	958.36	0.49	0.65	0.57	305.68
	ECRS	0.21	0.54	0.37	1181.91	0.51	0.67	0.59	296.68
	RRS	0.28	0.46	0.37	1082.33	0.39	0.53	0.46	365.26
	R-INSGA	0.44	0.69	**0.56**	838.88	0.64	0.73	**0.69**	213.56
Deliberate attack 2 (D5, S8, S7, D4, S3, D3, D9, S2, D7, D2, D6, I13, I16, D10, I19)	PBRS	0.19	0.63	0.41	1217.25	0.42	0.70	0.69	349.18
	TBRS	0.46	0.66	0.56	806.55	0.63	0.76	0.69	221.00
	DCRS	0.39	0.65	0.52	916.46	0.39	0.67	0.57	364.95
	BCRS	0.38	0.64	0.51	927.49	0.47	0.67	0.57	317.81
	CCRS	0.39	0.63	0.51	908.99	0.45	0.68	0.54	328.30
	ECRS	0.19	0.52	0.35	1217.25	0.42	0.67	0.56	349.18
	RRS	0.31	0.55	0.43	1040.23	0.39	0.58	0.48	365.10
	R-INSGA	0.45	0.70	**0.58**	820.64	0.64	0.79	**0.71**	218.22

direction. In addition, the initial positions of recovery teams significantly impact the recovery sequence of subsequent damaged entities. Scientifically determining the optimal initial positions for one or multiple teams presents a significant challenge for further exploration.

Electronic Supplementary Material Supplementary material is available in the online version of this article at 10.1007/s42524-025-4179-y and is accessible for authorized users.

Competing Interests The authors declare that they have no competing interests.

References

Artime O, Grassia M, De Domenico M, Gleeson J P, Makse H A, Mangioni G, Perc M, Radicchi F (2024). Robustness and resilience of complex networks. Nature Reviews. Physics, 6(2): 114–131

Aslani B, Mohebbi S, Oughton E (2024). A systematic review of optimization methods for recovery planning in cyber–physical infrastruc-ture networks: Current state and future trends. Computers & Industrial Engineering, 192: 110224

Chatterjee A, Helbig C, Malak R, Layton A (2023). A comparison of graph-theoretic approaches for resilient system of systems design. Journal of Computing and Information Science in Engineering, 23(3): 030906

Chatterjee A, Malak R, Layton A (2022). Ecology-inspired resilient and affordable system of systems using degree of system order. Systems Engineering, 25(1): 3–18

Chen W, Li W, Zhang T (2024a). Complex network-based resilience capability assessment for a combat system of systems. Systems, 12(1): 31

Chen Z, Hong D, Cui W, Xue W, Wang Y, Zhong J (2023a). Resilience evaluation and optimal design for weapon system of systems with dynamic reconfiguration. Reliability Engineering & System Safety, 237: 109409

Chen Z, Yin S, Li L, Cui W, Hong D (2024b). Resilience metric and dynamic assessment of unmanned system-of-systems considering cooperative reconfiguration strategies. IEEE Transactions on Reliability. doi:10.1109/TR.2024.3438810

Chen Z, Zhou Z, Zhang L, Cui C, Zhong J (2023b). Mission reliability modeling and evaluation for reconfigurable unmanned weapon system-of-systems based on effective operation loop. Journal of Systems Engineering and Electronics, 34(3): 588–597

Cimellaro G P, Reinhorn A M, Bruneau M (2010). Framework for analytical quantification of disaster resilience. Engineering Structures, 32(11): 3639–3649

Danziger M M, Barabási A L (2022). Recovery coupling in multilayer networks. Nature Communications, 13(1): 955

De Domenico M (2023). More is different in real-world multilayer networks. Nature Physics, 19(9): 1247–1262

Deb K, Pratap A, Agarwal S, Meyarivan T (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation, 6(2): 182–197

Deller S, Bowling S R, Rabadi G A, Tolk A, Bell M I (2009). Applying the information age combat model: quantitative analysis of network centric operations. The international C2 Journal, 3(1)

Deptula D A, Penney H R, Stutzriem L A, Gunzinger M (2019). Restoring america's military competitiveness: Mosaic warfare. Mitchell Institute for Airpower Studies

Du C, Ouyang M, Zhang H, Wang B, Wang N (2023). Resilience patterns of urban road networks under the worst-case localized

disruptions. Risk Analysis, 44(10): 2333–2347

Dui H, Zhu Y, Tao J (2024). Multi-phased resilience methodology of urban sewage treatment network based on the phase and node recovery importance in IoT. Reliability Engineering & System Safety, 247: 110130

Edwards C M, Nilchiani R R, Miller I M (2024). Impact of graph energy on a measurement of resilience for tipping points in complex systems. Systems Engineering, 27(4): 745–758

Fang Z, Wu S, Zhang X, Sun Y (2021). ADC-GERT network parameter estimation model for mission effectiveness of joint operation system. Journal of Systems Engineering and Electronics, 32(6): 1394–1406

Filippini R, Silva A (2014). A modeling framework for the resilience analysis of networked systems-of-systems based on functional dependencies. Reliability Engineering & System Safety, 125: 82–91

Gao F, He W, Bi W (2024). An intuitionistic fuzzy weighted influence non-linear gauge system for equipment evaluation under system-of-systems warfare environment. Expert Systems with Applications, 238: 122187

Gao Y, Cheng J, Tian Y, Liu H (2023). Machine learning-based evaluation of the contribution effectiveness in SoS missions. IEEE Systems Journal, 17(4): 5877–5888

Hafsa M, Wattebled P, Jacques J, Jourdan L (2025). Solving a multiobjective professional timetabling problem using evolutionary algorithms at mandarine academy. International Transactions in Operational Research, 32(1): 244–269

Han Q, Pang B, Li S, Li N, Guo P S, Fan C L, Li W M (2023). Evaluation method and optimization strategies of resilience for air and space defense system of systems based on kill network theory and improved self-information quantity. Defence Technology, 21: 219–239

Holling C S (1973). Resilience and stability of ecological systems. Annual Review of Ecology and Systematics, 4(1): 1–23

Huang Y, Luo A, Chen T, Zhang M, Ren B, Song Y (2023). When architecture meets RL+EA: A hybrid intelligent optimization approach for selecting combat system-of-systems architecture. Advanced Engineering Informatics, 58: 102209

Khasawneh M T, Shearer N, Rabadi G, Bowling S (2017). The information age combat model: A vision for a discrete event simulation approach. International Journal of Simulation and Process Modelling, 12(5): 429–445

Koehler M T, Bricio-Neto J L, Page E H, Tolk A (2024). Applying complex adaptive systems research results to combat simulations of the generation-after-next. The Journal of Defense Modeling and Simulation, 15485129241233608

Lanier B, Petnga L (2019). Spatial functions for modeling and analysis of safety-critical systems of systems. 2019 14th Annual Conference System of Systems Engineering, SoSE 2019, page 352–357

Li J, Jiang J, Yang K, Chen Y (2019). Research on functional robustness of heterogeneous combat networks. IEEE Systems Journal, 13(2): 1487–1495

Li J, Tan Y, Yang K, Zhang X, Ge B (2017). Structural robustness of combat networks of weapon system-of-systems based on the operation loop. International Journal of Systems Science, 48(3): 659–674

Li J, Zhao D, Jiang J, Yang K, Chen Y (2021). Capability oriented equipment contribution analysis in temporal combat networks. IEEE Transactions on Systems, Man, and Cybernetics. Systems, 51(2): 696–704

Li R, Yuan H, Ren B, Zhang X, Chen T, Luo X (2024). Optimal unmanned combat system-of-systems reconstruction strategy with heterogeneous cost via deep reinforcement learning. Mathematics, 12(10): 1476

Liu J, Xu R, Li J, Yang K, Lou Z (2024). Enhancing the resilience of combat system-of-systems under continuous attacks: Novel index and reinforcement learning-based protection optimization. Expert Systems with Applications, 251: 123912

Liu M, Li J, Yang Z, Yang K (2023). Higher-order functional structure exploration in heterogeneous combat network based on operational motif spectral clustering. IEEE Systems Journal, 17(3): 4279–4290

Liu X, Li D, Ma M, Szymanski B K, Stanley H E, Gao J (2022). Network resilience. Physics Reports, 971: 1–108

Lozano-Osorio I, Sánchez-Oro J, López-Sánchez A, Duarte A (2025). A variable neighborhood search for the median location problem with interconnected facilities. International Transactions in Operational Research, 32(1): 69–89

Markina-Khusid A, Jacobs R B, Antul L, Cho L, Tran H T (2022). A complex network framework for validated assessments of systems of systems robustness. IEEE Systems Journal, 16(1): 1092–1102

Miller K, Bordetsky A, Mun J, Maule R, Pollman A (2021). Merging future knowledgebase system of systems with artificial intelligence/machine learning engines to maximize reliability and availability for decision support. Military Operations Research, 26(4): 77–93

Qi X, Mei G (2024). Network resilience: Definitions, approaches, and applications. Journal of King Saud University – Computer and Information Sciences, 36(1): 101882.

Rescia A, Gómez Menéndez A, González Lodares C, Ortega M (2023). Quantification of the spatial resilience of the spanish rural landscape to fire occurrence analysed using the sispares network of plots. Landscape Ecology, 38(12): 3621–3636

Sinex C H, Lilly T C, Harlow M A (2000). Using the war room process to explore network-centric warfare. Johns Hopkins APL Technical Digest, 21(3): 368–377

Sun L, Zhou Y, Zhu C, Zhang W (2024a). A new important nodes identification method in multi-layer heterogeneous combat network with meta-path centrality. Journal of Complex Networks, 12(3): cnae021

Sun Q, Li H, Wang Y, Zhang Y (2022). Multi-swarm-based cooperative reconfiguration model for resilient unmanned weapon system-of-systems. Reliability Engineering & System Safety, 222: 108426

Sun Q, Li H, Zhong Y, Ren K, Zhang Y (2024b). Deep reinforcement learning-based resilience enhancement strategy of unmanned weapon system-of-systems under inevitable interferences. Reliability Engineering & System Safety, 242: 109749

Sun Y, Zhang T (2023). Research on autonomous reconstruction method for dependent combat networks. IEEE Systems Journal, 17(4): 6104–6113

Tran H T, Domerçant J C, Mavris D N (2016). A network-based cost comparison of resilient and robust system-of-systems. Procedia Computer Science, 95: 126–133

Uday P, Chandrahasa R, Marais K (2019). System importance

measures: Definitions and application to system-of-systems analysis. Reliability Engineering & System Safety, 191: 106582

Uday P, Marais K (2015). Designing resilient systems-of-systems: A survey of metrics, methods, and challenges. Systems Engineering, 18(5): 491–510

Wang H (2023). A new algorithm for euclidean shortest paths in the plane. Journal of the Association for Computing Machinery, 70(2): 11

Wang L, Jin J G, Sun L, Lee D H (2024a). Urban rail transit disruption management: Research progress and future directions. Frontiers of Engineering Management, 11(1): 79–91

Wang Z, Su Z, Deng Y, Kurths J, Wu J (2024b). Spatial network disintegration based on kernel density estimation. Reliability Engineering & System Safety, 245: 110005

Watson B, Chowdhry A, Weissburg M, Bras B (2022). A new resilience metric to compare system of systems architecture. IEEE Systems Journal, 16(2): 2056–2067

Watson B C, Morris Z B, Weissburg M, Bras B (2023). System of system design-for-resilience heuristics derived from forestry case study variants. Reliability Engineering & System Safety, 229: 108807

Pan X, Wang H, Yang Y, Zhang G (2019). Resilience based importance measure analysis for SoS. Journal of Systems Engineering and Electronics, 30(5): 920–930

Xu R, Gong L, Xie J, Liu X, Yang K (2023a). Operation network link importance evaluation and recovery strategy based on equipment system-of-systems resilience. Journal of Systems Engineering and Electronics, 45(1): 139–147

Xu R, Liu J, Li J, Yang K, Zio E (2024). TSoSRA: A task-oriented resilience assessment framework for system-of-systems. Reliability Engineering & System Safety, 248: 110186

Xu R, Liu X, Cui D, Xie J, Gong L (2023b). An evaluation method of contribution rate based on fuzzy bayesian networks for equipment system-of-systems architecture. Journal of Systems Engineering and Electronics, 34(3): 574–587

Yabe T, Rao S C, Ukkusuri S V, Cutter S L (2022). Toward data-driven, dynamical complex systems approaches to disaster resilience. Proceedings of the National Academy of Sciences of the United States of America, 119(8): e2111997119

Yang D, Li Q, Zhu F, Cui H, Yi W, Qin J (2023). Parallel emergency management of incidents by integrating ooda and prea loops: The c2 mechanism and modes. IEEE Transactions on Systems, Man, and Cybernetics. Systems, 53(4): 2160–2172

Zhang J, Fang Z, Dong W, Zhang L (2024). Portfolios selection decision model for equipment system of systems considering development costs. Expert Systems with Applications, 246: 123235

Zhong Y, Li H, Sun Q, Huang Z, Zhang Y (2024). A kill chain optimization method for improving the resilience of unmanned combat system-of-systems. Chaos, Solitons, and Fractals, 181: 114685

Zobel C W (2011). Representing perceived tradeoffs in defining disaster resilience. Decision Support Systems, 50(2): 394–403